REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
	Final Report	10 August 2005 - 10-Mar-06

4. TITLE AND SUBTITLE

Autonomous agents for time series prediction

5a. CONTRACT NUMBER
FA8655-05-1-3048

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dr. Philip Palmer

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Surrey
Guildford GU2 7XH
United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 821 BOX 14
FPO 09421-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
SPC 05-3048

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking University of Surrey as follows: The Surrey Space Centre, primary developer of the Disaster Monitoring Constellation (DMC), a network of satellites that provides users global natural and man-made event monitoring, seeks to monitor space and/or terrestrial source data streams for identifying interest-event occurrences. For the purposes of this research, an event is defined as a significant interest item that occurs at a particular time and location, such as an individual volcano eruption, a flood, or a forest fire. During- and postevent detection can often be achieved through one of several change detection algorithms, however pre-event detection introduces an entirely different challenge. Successful pre-event detection involves comparing temporal data against unique impending event data patterns. More concisely, successful pre-event detection involves combining time series analysis with robust event pattern recognition. While domain-specific methodologies have garnered varying success levels, a general approach for this complex task has yet to be found and therefore motivates this research effort.
Significant progress across the range of research goals and objectives has been achieved. Preliminary analysis results using one and two channelled data suggest the method is capable of identifying complex event-related data patterns and perhaps even predicting significant events. These results strengthen our conviction the method warrants further research and investigation.

15. SUBJECT TERMS
Operations research, autonomous agent, time series prediction,EOARD

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			BARRETT A. FLAKE
UNCLAS	UNCLAS	UNCLAS	UL	54	19b. TELEPHONE NUMBER *(Include area code)* +44 (0)20 7514 4285

Near Real-Time Event Detection & Prediction Using Intelligent Software Agents

For Fulfilment of Grant #FA8655-05-1-3048

Dr. Phil Palmer

Surrey Space Centre
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

March 2006

Abstract

The Surrey Space Centre, primary developer of the Disaster Monitoring Constellation (DMC), a network of satellites that provides users global natural and man-made event monitoring, seeks to monitor space and/or terrestrial source data streams for identifying interest-event occurrences. For the purposes of this research, an event is defined as a *significant* interest item that occurs at a particular time and location, such as an individual volcano eruption, a flood, or a forest fire. During- and post-event detection can often be achieved through one of several change detection algorithms, however pre-event detection introduces an entirely different challenge. Successful pre-event detection involves comparing temporal data against unique impending event data patterns. More concisely, successful pre-event detection involves combining time series analysis with robust *event* pattern recognition. While domain-specific methodologies have garnered varying success levels, a general approach for this complex task has yet to be found and therefore motivates this research effort.

The overall research goal is to develop, test, and validate a robust generic methodology for determining if a complex process is undergoing, or about to undergo, a significant change. More specifically, this novel research aims to:

- Develop, test, and validate a generic, robust methodology for near real-time (pre-) event detection and monitoring on one or more data streams *without requiring access to domain-specific knowledge or analytical models*
- Apply this methodology to noisy and non-stationary raw data as its primary source of input, while minimizing all other data requirements
- Minimize required user interactions while allowing user insight into the training and prediction process

Within the context of this research grant, the major research milestones for this effort included conducting preliminary sensitivity analyses on the proposed methodology, such as:

- Evaluating method performance
- Evaluating variations of existing or different statistical features
- Evaluating best methods to monitor and control analysis processes
- Evaluating derived feature and methodology efficacy by incorporating real data from a monitored environmental process and comparing pattern recognition results against results produced by experts in the related environmental field

Significant progress across the range of research goals and objectives has been achieved. Preliminary analysis results using one and two channelled data suggest the method is capable of identifying complex event-related data patterns and perhaps even predicting significant events. These results strengthen our conviction the method warrants further research and investigation.

Contents

Chapter 1

Introduction

1.1 Overview of Surrey Space Centre Autonomy Effort

Current space systems usually involve sizable operations and support structures. While some mundane tasks relating to operating, maintaining, and tasking satellite constellations are performed automatically, most activities require continuous human interaction. The extent to which humans must interact on an hour-by-hour or day-by-day basis with the constellations likely results in less than optimal constellation performance and introduces additional delays into the overall process of responding to urgent data requests by users. If it were possible to limit, to the maximum extent possible, the need for humans to interact with the satellite constellations to perform functions that could be otherwise performed autonomously by the satellite constellation itself (either on the ground or on the actual satellites), then the potential exists for better constellation utilization and more timely responses to users' needs. Improved response times, however, further magnifies the growing global quandary of task saturation and data overload often experienced by users. This problem is especially acute for users confronted with the arduous task of monitoring and analysing one or more streams of complex process data from multiple sources in order to detect when significant events of interest have or are likely to occur. Accordingly, developing the capability to automatically perform this challenging (pre-) event detection task on complex time series data is of considerable interest.

The Surrey Space Centre (SSC) has taken up the challenge to improve upon existing practices and research how to best develop the concepts described above as part of a comprehensive autonomy effort. The overall aim of this effort is the development of an efficient process for requesting space related data, tasking space assets to meet data requests, delivering data products to users, and automatically analysing multiple data streams for (pre-) event detection. The SSC identified five distinct layers that, when properly put together, construct this comprehensive system. These layers are shown in Figure 1.1, and the research effort described in this report aims to develop much of the first layer. A brief summary of the different layers follows.

The first layer of the small spacecraft autonomy solution represents the users. In this layer, users have access to regularly updated time series data from various sources (space or terrestrial) concerning a process of interest, and intelligent software agents integrate the raw data into finely tuned statistical features for use in near real-time event detection and prediction (a form of pattern recognition). Alerts are generated when the values of statistical features or data patterns match those previously found to be indicative of an impending (or occurring) event. The completed first layer would also include the capability for users to obtain data not presently accessible through other means by employing a different type of intelligent software agent, notionally called a contract agent, to contract with a data service provider to supply the missing data.

The second layer represents the peer-to-peer computer network(s) that connects the users to one another as well as to data service providers. The chosen network architecture offers users working in different domains a large degree of flexibility to rapidly share information and enter/leave the network as needed. Indexing of information available throughout the network is also considered within this layer.

The third layer of the small spacecraft autonomy solution, representing the space service provider, attempts to best accommodate all user data requests in the timeliest possible manner given the constraints of its space system constellation at any point in time. A key aspect of this research is the generation of statistical information allowing users to understand the likelihood of receiving desired space data within their desired timeframe. The near real-time event detection and prediction method

may also be applied within this layer to automatically alert space service providers of impending significant events for possible (re-) tasking of space assets or task reprioritisation as necessary.

The fourth layer lies within a single satellite and optimizes the spacecraft's resources to best meet the tasks placed upon it by either the space service provider or the virtual satellite (representing the fifth layer). The fourth layer must resolve the conflicting resource issues caused by users' requests and spacecraft subsystem needs.

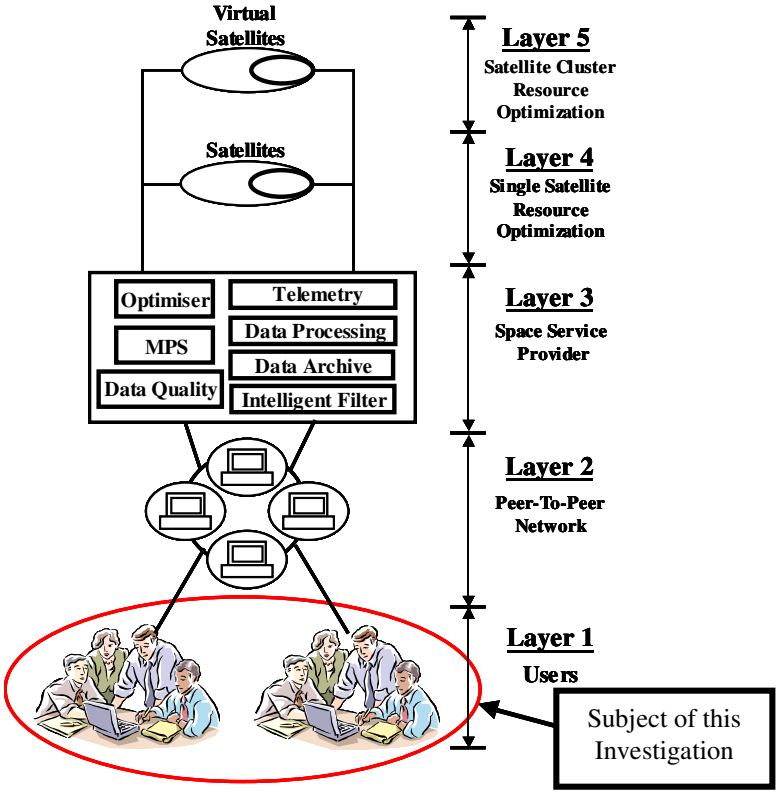

Figure 1.1. SSC's Multi-Layered Approach to Solving the Autonomy Challenge

Finally, the fifth layer of the small spacecraft autonomy taxonomy investigates the potential of a cluster of smaller satellites flying in formation to effectively operate as a single larger "virtual satellite." The goal of the virtual satellite construct is to minimize network administration traffic overhead while maximizing the use of the various payloads contained within the cluster to best meet given sets of tasks delivered by the ground station(s).

The total research effort outlined above investigates key elements that must be improved for space systems to be more efficient for space developers, operators, and users of information products. While the investigations involve analysing components of satellite systems and processes developed over years through the SSC and Surrey Satellite Technology Limited (SSTL) partnership, the methods for improvement are intentionally generic in nature so as to be broadly applicable.

1.2 Overview of Research Effort

Event detection, time series analysis, and pattern recognition are all broad topics with a vast literary base. Over the years, numerous researchers have combined one or more of these concepts, resulting in volumes of literature containing time series data analysed for either event detection or pattern recognition. Many methods employ data pre-processing routines prior to analyzing time series data,

resulting in smoother data or relevant statistical feature extraction. There are as many ways to approach these types of analyses as there are researchers interested in finding workable solutions.

Unfortunately, not all time series data analysis techniques have proven successful. Certainly there are many reasons for failure - only a few are mentioned here. One possible reason lies with the analysed data's complexity. This occurs with very noisy data and/or data exhibiting highly non-stationary characteristics (changes in mean, variance, or both). Another reason includes mismatches between the analysis method and the nature of the available data. This can occur when the required analytical assumptions are not properly met. For those methods found to be successful, few thrive in multiple fields of research. One example of a very successful method with broad application is statistical process control (SPC). While promising new techniques to increase its effectiveness have been combined into the SPC framework, the basic concept has remained unchanged. The heart of SPC lies in the notion that a monitored process can be found in-control or out-of-control by comparing the magnitude of the difference between a metric computed using recent process data against the same metric computed over a longer time period. Looking beyond time series analysis, two more examples of successful methods capable of solving broad ranges of challenging problems across multiple fields of research are neural networks and genetic algorithms. These methods have largely succeeded due to inherent flexibility and relative applicability ease. The approach taken here combines these and other successful broad-based techniques, either directly or in modified form, to produce an altogether novel and generic methodology applicable to numerous disciplines.

1.3 Research Goal and Aims

The overall research goal is to develop, test, and validate a robust generic methodology for determining if a complex process is undergoing, or about to undergo, a significant change. More specifically, this novel research aims to:

- Develop, test, and validate a generic, robust methodology for near real-time (pre-) event detection and monitoring on one or more data streams *without requiring access to domain-specific knowledge or analytical models*
- Apply this methodology to noisy and non-stationary raw data as its primary source of input, while minimizing all other data requirements
- Minimize required user interactions while allowing user insight into the training and prediction process

1.4 Research Objectives

To accomplish the research aims, the following objectives have been established:

- Develop easy to use, continuous data collection and management process for one or more data streams within an intelligent agent construct
- Develop method to generate highly tailored robust statistical features from time series data that best capture trend and volatility movement found prior to and during previous significant events, capable of overcoming difficulties caused by non-stationary data
- Develop method for monitoring, controlling, and configuring first-stage modified SPC tool
- Develop method for combining statistical features from multiple data streams and generating a composite "snapshot" of the temporal process
- Develop method to adaptively configure second-stage pattern recognition tool (a neural network)
- Develop method for intelligent agent to interactively control overall (pre-) event detection analytical process
- Perform sensitivity analyses on methodology using synthetic and real data

1.5 Research Novelty

This research effort is novel in several areas, summarized below.

First, the methodology described in this report offers the potential to overcome some of the difficult challenges experienced by researchers analysing time series data for event detection by combining a modified form of SPC and pattern recognition into a single process. This modified SPC approach is powerful in two key areas:

- It generates robust statistical features tailored to the actual movement of complex time series data prior to and during previous significant events
- It simultaneously generates optimal control limits (or in this application optimal alert notification limits) for those statistical features

By simultaneously solving for both the features and the control limits, the methodology optimally captures and characterises how the time series evolved prior to and during previous significant events. This integrated process effectively defines a data movement pattern that can then be matched against future data as the time series evolves temporally. No other approaches found within the literature directly combine these two aspects and in fact, finding SPC-like methods applied to other than controlled industrial processes is rare indeed. There is little doubt monitoring of uncontrolled processes with standard SPC methods has proven challenging. Zimmerman [107], when discussing the difficulties associated with monitoring water pollution using standard SPC and time series techniques, listed some common problems: seasonal variation, high natural variability, lack of independence, covariate effects, non-normal distributions, auto correlation, and unreliable data. The methodology employed here does not require making the common assumptions used in SPC analyses, and consequently is not affected by many of these difficulties. In fact, should some of these "problems," such as high natural variability, be somehow related to the events of interest temporally, then these "problems" may actually aide the algorithm in discovering relevant data patterns. With other pattern recognition or point estimating analysis techniques where raw data is first pre-processed for feature extraction or smoothing and then analysed serially, there is no way to ensure an optimal match occurs between the prepared data and the chosen analysis method.

Second, the method of derived feature development and the nature of the statistical features being developed enable characterization of highly complex time series data without first requiring the raw data to meet strict SPC assumptions of normality and stationarity. Discussed more fully later, the types of statistical characteristics generated through the proposed method include moments and volatility metrics (such as standard errors) over varying time windows, metrics not dependent on distributional assumptions or data stationarity. Rigorously meeting these assumptions, while desirable, is not necessary since the goal of the analytical method is pattern recognition and not point prediction or hypothesis testing (as in the case of standard SPC applications). This means the statistical metrics can be calculated from the raw temporal data directly without needing first to resolve common time series data analysis problems as required by most analysis methods. In fact, the author was unable to locate other instances where SPC (or SPC-like) methods were applied to data still within the time domain (such as the aforementioned features) without having first to meet the general SPC assumptions of sample data normality and stationarity.

Third, the methodology includes the concept of intelligent software agents within an event detection algorithm. Multiple agents are used within the research methodology: retrieving raw data, monitoring and executing the pattern recognition tasks, and supervising the overall analytical process. Employing the intelligent software agents effectively eliminates much of the continuous interaction normally required by users when analysing complex data in any domain. While the concept of an agent is certainly not new, other instances where an agent has controlled an SPC-like application for (pre-) event detection were not discovered in the literature.

Chapter 2

Literature Review

2.1 Overview of Literature Review

There are a vast number of ways to analyze time series data, depending on the purpose of the analysis. Sections 2.2 and 2.3 review how controlled and uncontrolled processes can be monitored for (pre-) event detection. In the case of controlled processes, the relevant basics of static and adaptive statistical process control are addressed, including several discussions about state-of-the-art practices where multiple methods are combined. In the case of uncontrolled processes, reviewing process-unique models is not as constructive as introducing common modelling techniques presently used by researchers. To keep this task manageable, the review will focus on those techniques used by researchers within the environmental monitoring field. Since most monitoring methods often require the development of statistical features derived from raw process data, this topic will be briefly mentioned throughout the sections. Sections 2.4 and 2.5 briefly introduce to two of the most versatile and widely-used pattern recognition and optimization techniques: neural networks and genetic algorithms. Additional discussion about genetic algorithms can also be found in Section 3.3.

2.2 Overview of Static and Adaptive Statistical Process Control

The concept of statistical process control (SPC) is based on the idea that a process, such as a manufacturing process, can be monitored to identify when it is "in control" or "out of control." Monitoring is accomplished through product sampling, and adjustments are made to the production process as needed in an attempt to continuously reduce the number of defective products produced. In the 1920's Walter Shewhart described this concept [76], the basic philosophy of modern SPC, and thus started the movement now commonly referred to as "total quality management."

In SPC, process variation is measured and broken down into two components [1]. The first component is called common cause, or system variation. This type of variation is the naturally occurring fluctuation, or variation, inherent in all processes. Making modifications to the process can reduce system variation. The second component is called special cause variation, meaning the variation is due to some type of extraordinary occurrence in the production process. These tend to be localized in nature and are often attributed to causes such as operator error, machine error, or materiel inconsistencies. Special cause variation is the foremost type of variation Shewhart's methods attempted to identify and correct, since this variation leads to "out of control" processes. Over time, however, both types of variations are hopefully reduced through continual monitoring and process improvement, with the result being overall improved product quality and reduced sample-to-sample variation. To accomplish this, Shewhart developed a methodology for charting a process to quickly determine the current state of a system. While different variations of the basic SPC chart have evolved over time, the basic chart construct remains the same. Samples are taken over time and plotted on a time-series control chart. The values are compared to long-running characteristics, such as the sample mean (known as an $\overline{X}$-chart), and its position relative to upper and lower control limits is identified. The control limits identify when the process goes "out of control." In the case of the $\overline{X}$-chart, they are defined as the mean, μ, plus or minus the control limit coefficients, k, times the process standard deviation, σ, divided by the square root of the sample size, n [76]. A frequently used value for k is three and the equations for $\overline{X}$-chart upper and lower control limits (*UCL* and *LCL*, respectively) are given in Equations 2.1 and 2.2 [45].

$$UCL = \mu + \frac{k\sigma}{\sqrt{n}} \qquad \text{Equation 2.1}$$

$$LCL = \mu - \frac{k\sigma}{\sqrt{n}} \qquad \text{Equation 2.2}$$

Figure 2.1 shows an example of a process apparently in control, due to all samples falling within the upper and lower control limits. The variation exhibited in this figure represents naturally occurring variation.

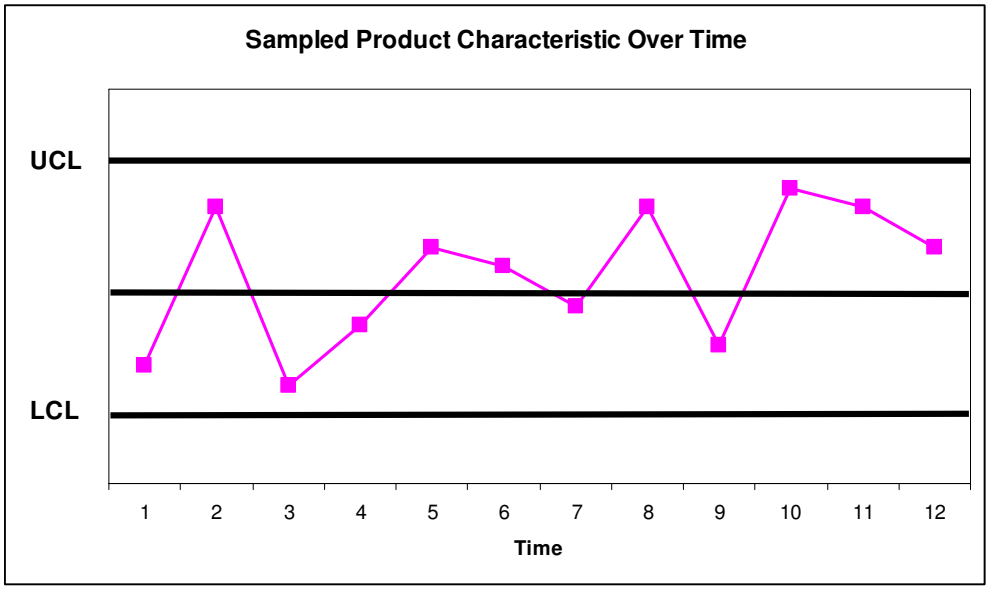

Figure 2.1. A Process Apparently In Control

Had an observation value fallen outside one of the limits, then SPC theory states special cause variation was present. To increase awareness of excessive variations or variations that change with time, SPC practitioners often include one or more zones between the upper and lower control limits, and track how data moves through the different zones [31]. Identifying patterns of movement, such as linear trends, oscillations, or numbers of points falling into certain zones represent early detection methods for processes before they go "out of control."

There are a variety of SPC charts, depending on whether or not the data is continuous or discrete. Commonly used charts include [76]:

- For *continuous* (variables) data
  - Shewhart sample mean chart ($\overline{X}$-chart): plots raw variables, x_i, compared to the sample mean over time
  - Shewhart sample range chart (*R*-chart): plots ranges observed within a sample run
  - Shewhart sample chart (*X*-chart): plots the raw variables, x_i, over time
  - Cumulative sum chart (CUSUM): plots raw variable values, x_i, or the difference between x_i and a target, its mean, or the previous value, x_{i-1}, over time
  - Exponentially Weighted Moving Average (EWMA) chart: plots an exponentially weighted moving average over time
  - Moving-average and range charts: plots the simple moving average or sample range over time

- For *discrete* attributes and countable data, common charts address defect proportions and rates, topics outside the scope of this research effort.

A primary function of SPC is to be able to decide whether or not the monitored process is statistically stable. To accomplish this task, SPC has historically been based on two key assumptions concerning the plotted statistic:

- It is *independent*, i.e. the value is not influenced by its past value and will not affect future values
- It is *normally* distributed, i.e. the data has a normal probability density function

Accordingly, SPC theory considers control charts to be plots of samples drawn from normal distributions over time, as shown in Figure 2.2 [28].

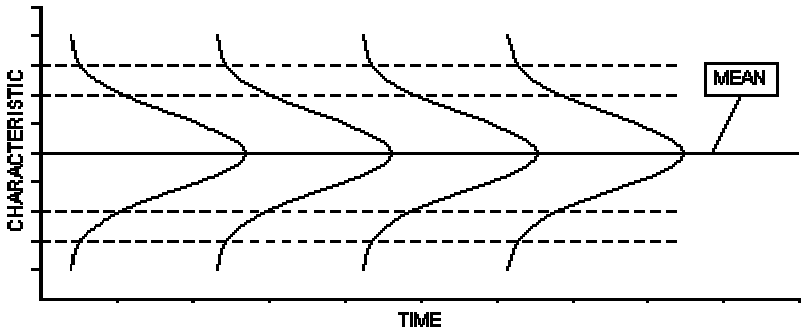

Figure 2.2. Control Charts Assume Sampling from Normal Distribution Over Time

The reason *k* is frequently set to three (often the case for critical product characteristics) is due to the normality assumption and SPC's form of hypothesis testing. Given the probability of randomly drawing a sample value 3σ or more away from the mean, μ, of a normal probability density function is approximately 0.002, then it is very likely an observed sample falling outside the *LCL* or *UCL* defined by μ +/- 3σ is due to a source of special cause (assignable) variation rather than natural variation and the process is termed "out of control." Given that processes invariably change over time (due to improved processes and reduced volatility, gradual machine wear causing increased natural variation, etc.), occasionally it is necessary to re-establish the "fixed" control limits while holding other SPC parameters constant [56].

From the mid-1990's onwards, one finds a noticeable increase in the volume of SPC literature detailing adaptive or dynamic constructs in SPC operations [31][56][87]. The goal of identifying ever smaller magnitude shifts in monitored characteristics focused attention on the need for changing one or more chart parameters in real-time based upon the observed sample values *under the assumption of normality*. In addition, continuing challenges to the normality assumption (due to the frequent practice of using smaller sample sizes than necessary to invoke the central limit theorem) drove the need to explore the affects of samples drawn from non-normal distributions [56].

Adaptive SPC introduced the opportunity to change one or more of the following common chart parameters: sample size, *n*; time between taking samples, *h*; and control limit coefficients, *k*. If additional zones have been identified between the *UCL* and *LCL*, such as "warning" zones, then warning limit coefficients *w* (synonymous with control limit coefficients) can also be changed. For moving average charts (either simple or exponentially weighted), additional adaptive parameters include the moving average length, *l*, and the smoothing constant, λ. Changes in the parameter values are based on the perceived stability of the process following the most recent value of the monitored

sample characteristic. Typically, two values for each parameter exist, one larger or longer than the other. If the process is perceived to be statistically stable, then changes such as decreasing n and λ, or increasing k, h, w, and l can be considered [31][56][87]. Changing the same parameters in the opposing directions is frequently considered should the process be perceived as becoming less statistically stable. Adaptive SPC naturally led to combining SPC with other analytical methods in an effort to further enhance sensitivity.

Early detection of problems can often be achieved by combining SPC with artificial intelligence methods, notably neural networks, resulting in improved process control. Many researchers found combining neural networks with certain statistical features within an SPC construct particularly effective. [1] trained a neural network on the standard control chart patterns and extracted multi-resolution wavelets from the process data for monitoring process control. The researcher found the neural network was better able to adjust to the actual process data movement than the fixed zone methods frequently used in static control charts because the complex time-frequency characteristics of non-stationary data in control charts is too much of a challenge for static control charts to handle. In addition, he found the use of fixed window frequency analysis techniques, such as the short term Fourier transform, produce high-frequency localization results with poor time resolution, whereas high time resolution results in poor frequency localization. [2] used an unsupervised neural network within the research to good effect, however it is important to note the researcher's methods assumed data normality. Despite this assumption, the researcher explored training the neural network not on pre-defined data patterns representing process irregularities but rather on the actual process data. [40] investigated using neural networks with non-normal data within control charts. The researcher's conclusion was despite many industrial processes not producing samples meeting the normality assumption, neural networks were still able to perform the pattern recognition task. Different neural network architectures and configuration have been investigated and found successful, with the most popular being back propagation [108], and recurrent neural networks [76]. Many of these researchers only monitored single data streams, so a multivariate perspective is also necessary.

Multivariate SPC (MSPC), where multiple data streams are analyzed simultaneously, have also been employed with and without neural networks. [88] successfully combined multivariate SPC with neural networks after the usual linear analysis techniques commonly employed in MSPC failed. The researchers found the nonlinear relationships between the process parameters could not be discovered through principle component analysis or partial least square methods, but they could be identified by employing a neural network. Other researchers of MSPC, such as [108], successfully employed neural networks and MSPC for novelty detection. Their work monitored the process mean vector, based on probability density estimates and a Gaussian log-likelihood threshold.

Excellent discussions about SPC theory, how it operates, its sensitivities to sampling distributions, and other topics can be found by reading any number of publications and books authored by reference sources, notably those written by Montgomery. Details about many of these topics, while interesting, rapidly fall beyond the scope of the research effort and its use of SPC concepts.

2.3 Overview of Environmental Monitoring Techniques

Researchers monitoring environmental processes tend to develop and use process-unique approaches. A review of some common elements, however, is still useful within the context of this research.

While few researchers use SPC methods to monitor the environment, several articles in the literature were found where this was attempted, but not always successfully. [28] performed emission monitoring from an industrial process using process capability indices with a cumulative sum chart. They accomplished this task with only slight modifications to the basic SPC control limit equations. The environmental data required substantial pre-processing, and they evaluated a range of transformations to make observations behave normally for use within the standard SPC construct.

Their transformations included logarithms, inverses, square roots, and others. Another example of environmental monitoring with SPC methods, albeit in a relatively controlled environment, is found in [46]. They monitored early warning fire detection systems for the US Navy and found MSPC could be applied successfully using Hotelling's statistic and Q-statistic following principal component analysis. As mentioned earlier, however, other researchers have found these methods perform poorly as the data becomes more complex. [107] used SPC to monitor water quality of a bay and found SPC techniques unable to overcome the extreme difficulties of data not conforming to standard SPC assumptions. In addition, the researcher identified a need to develop a much more rigorous method of data collection since the observational data came from volunteers at irregular intervals.

A range of techniques outside SPC are regularly used for monitoring environmental processes. Neural networks continually reappear as a common technique [3][5][9][68][78][89][94]. [3] added auxiliary data to a water stream flow forecasting model, suggesting perhaps data not originally thought to be important might ultimately be relevant. [9] found a recurrent neural network outperformed existing atmospheric and time series models when modelling wind speed and power. [68] attempted to model SO_2 pollution with various methods, including statistical techniques and neural networks, and found none performed very well.

Most researchers attempted various transformations on the data before analysing it with their preferred methods. The degree to which researchers discussed their use of data transformations and data pre-processing varied widely, however. Besides those already mentioned, additional data transformations found in the literature include detrended fluctuation analysis [29] used to successfully analyse volcano data exhibiting high variability, a range of linear techniques (variants of autoregressive parameterization), Fourier techniques, non-linear techniques, and neural networks [5]. While some form of data transformation or data pre-processing often improves analysis results, some researchers found otherwise. For example, the researchers monitoring river flow data [5] found leaving the data raw (with no pre-processing) and analysing it with a neural network produced better results than pre-processing it and analysing with the neural network or other methods.

2.4 Overview of Neural Networks

Artificial neural networks (ANNs) are inspired by nature and loosely based on how scientists believe brains function and organisms learn. It is well understood that the brain is composed of a network of interconnected neurons. Neurons receive simultaneous inputs from other neurons through their dendrites, causing some neurons to "fire" as they pass or suppress signals along the network [13]. The firing of various neurons, along with a changing network structure and weighting of the respective neurons, forms the basis for how organisms learn. This same concept of a network, including neurons connected to each other and interacting with one another simultaneously, is the structure and learning principle used in ANNs. Learning is accomplished by providing feedback to the network under supervised training to adjust the model parameters in order to provide more accurate model output.

Early users of ANNs, such as McCulloch and Pitts in 1943, created simple networks that involved neurons firing only when summed inputs exceeded bias threshold values [13]. In the 1950's, Rosenblatt challenged the models made by McCulloch and Pitts because they were single layer in nature, didn't take into account randomness inherent in many systems, and therefore only had limited capabilities and uses. His ideas led to the development of Rosenblatt's perceptron, shown in Figure 2.3.

Rosenblatt's perceptron creates essentially a two-layer network (the input layer is not counted). The first layer contains fixed threshold logic functions, and the second layer provides the network output and has connecting trainable weights, w_J. Rosenblatt's perceptron improved an ANN's ability to distinguish between linearly separable functions, thus allowing it to perform adequately as a simple classification system. It still fell short, however, of accurately classifying regions that were not linearly separable, such as Exclusive OR (XOR) classification problems shown in Figure 2.4. In this

case, the existing learning algorithms will never terminate, and any arbitrary stopping rules do not guarantee that the resulting weight vector from the network will generalize well for new data.

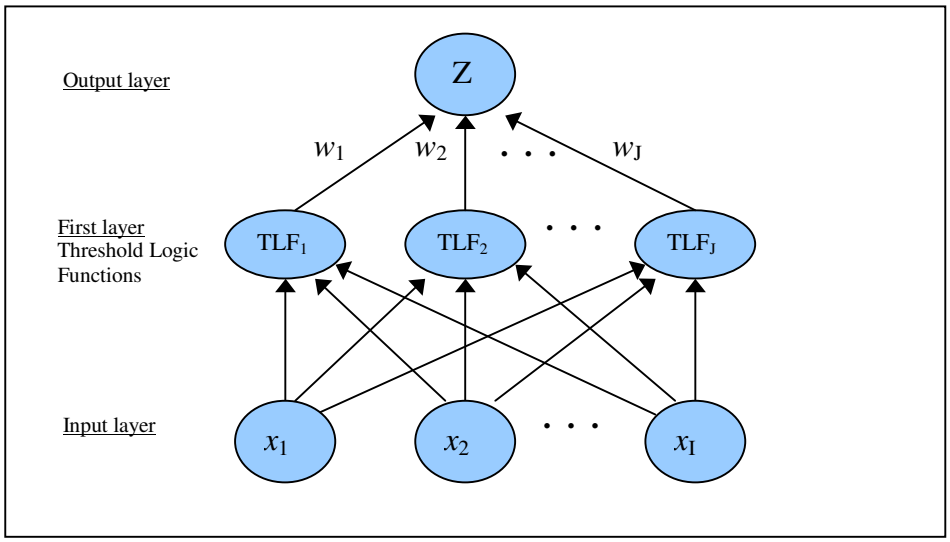

Figure 2.3. Rosenblatt's Perceptron

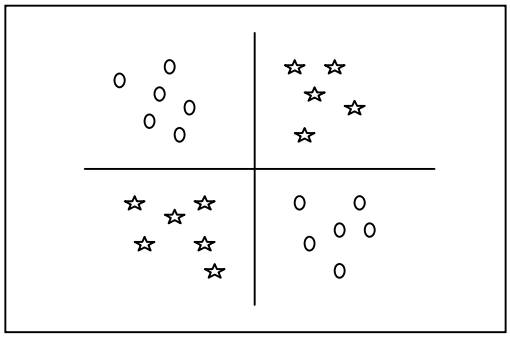

Figure 2.4. XOR Classification Problem

Minsky and Papert pointed out in 1969 that the reason these perceptron networks failed to correctly classify data sets that are linearly inseparable is due to the network structure only having a single layer of weights that are modified by the learning algorithm [72]. They showed that a network could solve a multi-dimensional problem, such as the XOR problem, as long as the number of perceptrons increased exponentially with the dimensionality of the problem being presented to the network. This would allow the ANN to operate in a transformed space where the problem can once again become linearly separable. Despite this discovery, size and computational limitations lead most researchers to believe that ANNs had little practical use for everyday problems and little progress was made toward improved learning algorithms or network structures until the late 1980's.

In 1986, Rumelhart, Hinton, and Williams announced the discovery of a new learning algorithm that eliminated the need for an exponential number of perceptrons to solve nonlinearly separable problems. Their approach, now called backpropagation, revitalized the ANN community by employing a gradient search method on the error surface produced following training. The gradient search method is implemented to minimize the error so the network correctly classifies patterns as often as possible. Other modifications to backpropagation have been introduced since the late 1980's, but the backpropagation method has remained the most widely used neural network algorithm by researchers and practitioners alike. A diagram is shown in Figure 2.5 with a single hidden layer and a bias term. $w^1_{i,j}$ represents a first-layer trainable weight between input node i and hidden node j, and $w^2_{j,k}$

represents a second-layer trainable weight between hidden node *j* and output node *k*.

In order for an ANN to be useful in classifying exemplars, the network must first be trained. Training any neural network involves an iterative process by which the network receives inputs, pumps them through the network using the current weight values, calculates the network outputs and the resulting error values based on comparisons with the known outputs, and then modifies the various nodal weights throughout the network in efforts of reducing the calculated error.

The backpropagation method is one algorithm of several to update the weights within a network. The cornerstone of the backpropagation algorithm lies in differentiable activation functions. This is important because the activations of the output nodes become differentiable functions of both the input variables, and of the weights and biases. If an error function is applied, such as a sum-of-squares error function, a differentiable function of the network output is created and the error is a differentiable function of the weights [13]. By evaluating the derivatives of the error function with respect to the different weights, weight values that minimize the error can be found. The backpropagation algorithm evaluates these derivatives and updates the various using a gradient descent approach to find the minimum error on the error surface. Detailed descriptions of the algorithm can be found in numerous texts, notably [13] and [72].

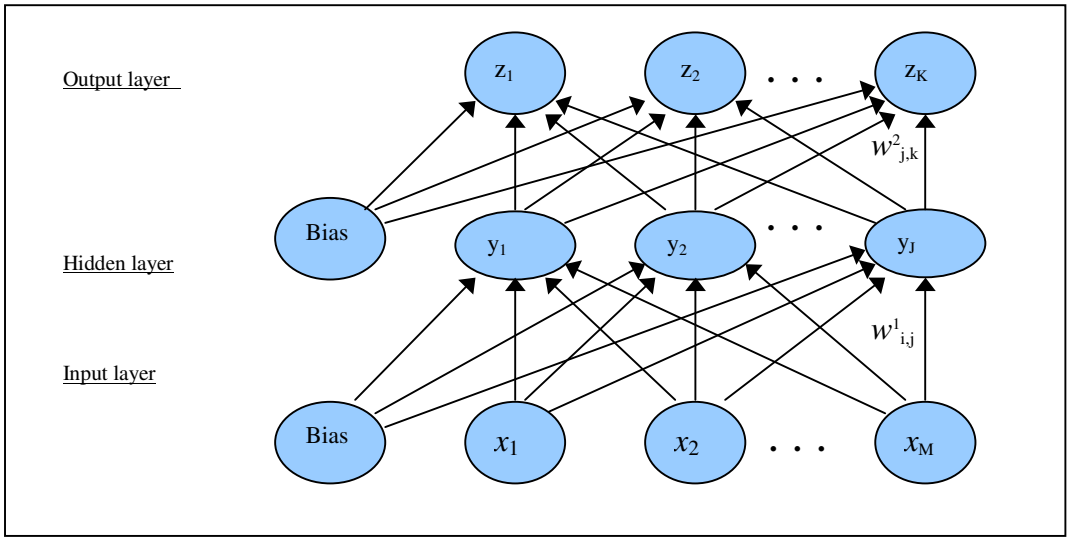

Figure 2.5. Multivariate MLP ANN with Bias Term.

2.5 Brief Overview of Genetic Algorithms

Genetic algorithms are also inspired by evolution and operate on the concept of simultaneously processing schemata from a gene pool [74]. Each member of the population consists of a number of genes, and a fitness function is applied to determine the relative fitness of each member. Following evaluation, candidate schemata are selected for the next generation in proportion to their relative fitness. Each pair of selected chromosomes is crossed with probability p_c, and some are mutated with probability p_m. This process is repeated generation after generation until a stopping condition is met and the optimal (or near optimal) chromosome is identified. Crossover and mutation offer genetic algorithms the benefit of rapidly moving across a complex surface without regards to differentiability, a notable difference when compared to the requirements most analytical methods. This makes using a genetic algorithm for combinatorial problems, such as the one in this research, particularly appealing. Additional discussions about genetic algorithms and how they are applied in this research can be found in Section 3.3. Detailed discussions, theory, and additional applications can be found within numerous texts [27], [36], [37], [74] and other literature.

Chapter 3

Development of Event Pattern Recognition Algorithm

3.1 Algorithm Overview

SPC methods, despite their popularity with researchers within industries where processes are monitored and controlled, are rarely applied to situations where processes are monitored but not controlled. This observation is rather understandable, since control charts offer a way to test whether or not the monitored process is in a state of statistical control. Out-of-control situations, identified by process data falling outside pre-defined control limits, can be considered *events* to be avoided and the quantity of these events can often be reduced through early detection of process problems. By comparing process data against a set of pre-defined patterns found to precede out-of-control processes, corrective actions can be taken and out-of-control situations avoided. By contrast, however, researchers working in fields where processes are uncontrolled, such as natural processes encountered within the field of environmental monitoring, do not have methods of "correcting" processes when out-of-control situations occur or are predicted to occur. Instead of using a form of SPC for monitoring, researchers in these fields accomplish their objectives through the frequent practice of building process-unique models based on first principles (if possible), past observations, and/or a combination of both. Their monitoring objectives may consist of developing point estimates for processes in the near future, such as predicting outside temperatures for selected geographical locations, or their objectives may consist of trying to predict (or detect) significant changes or *events* within underlying processes. If their task is the latter, then despite differences in approach and the lack of corrective measures to re-establish control over processes, their objective of monitoring process data for *(pre-) event detection* is essentially the same objective as researchers working in industries with controlled processes. Both groups of researchers are interested in repetitively answering the question, "are our monitored processes about to go out-of-control?" If we accept this argument, then under these circumstances the distinctions between the concepts of *(pre-) event detection* and *pattern recognition* are no longer necessary; the data pattern(s) found to precede (or coincide with) significant events is (are) the pattern(s) to be recognized.

The algorithm within this research effort must be generic in nature yet capable of continuously retrieving complex raw data, generating highly tailored features capturing the "event state" at discrete points in time, identifying any repeatable data patterns that exist preceding (or coinciding with) significant events, and monitoring the data stream(s) for the reappearance of these critical patterns. Should any of these (pre-) event patterns be discovered, the algorithm must alert the user(s). In addition, the algorithm must not require access to domain-specific knowledge or analytical models for the monitored system, it must be applicable to noisy and non-stationary raw data as its primary source of input, and it must minimize required interactions with users.

Fortunately, the novel algorithm developed within this research to meet these demanding criteria is strikingly straightforward. It can be summarized in five basic steps, identified below, and broken into two information flow diagrams, Figures 3.1 and 3.2.

1. *Initialization*. The intelligent agent queries the user for one or more data streams to monitor, and asks for objective function values. The objective function values indicate how to score statistical observations (or "observations") that exceed limits. Equation 3.1 details the fitness function, with the objective of maximising the fitness score. The user is also queried about how to handle alert notifications.

    - If a new data stream is identified, the agent requests historical data and the timing of past events within the historical period. With historical data, the agent can identify the raw data observation numbers associated with each event. Without historical data

the agent saves the data stream information and looks for significant variations in the data using broad settings, querying the user for event confirmation as necessary. With several confirmed events, the agent can proceed to the statistical feature and warning limit optimization process.

- To evaluate the efficacy and robustness of the statistical feature settings, the historical data is split into two or more groups. The ideal case entails developing three data sets (a training set, a testing set, and a validation set), with each data set containing multiple historical events.

2. *Statistical feature and warning limit optimization process*. The method uses single and dual-layered genetic algorithms to simultaneously define settings producing optimal combinations of select statistical features and associated warning limits. This part of the methodology contains much of the research novelty.

    - To find the optimal combinations, the genetic algorithms freely adjust a range of parameters similar to those found in traditional SPC for charting monitored processes and score the result. The method imposes no constraints on how the warning limits are defined, allowing trend-based limits and constant-valued limits to be created and mixed interchangeably. Furthermore, by allowing the control limit coefficients, k, to range into negative territory, an event "band" can be created and evaluated. In this case, the lower warning limit rises above the upper warning limit, and observations within this region simultaneously exceed both warning limits. The parameter values producing the highest fitness score for each statistical feature and warning limit combination are maintained. The range of statistical features include: simple moving average, velocity, acceleration, volatility, and change in volatility.

    - The method accommodates single as well as joint conditional relationships. Solving a single conditional relationship entails finding the optimal settings for one statistical feature and warning limit combination for one data channel. Solving a joint conditional relationship entails simultaneously finding the optimal settings for two statistical feature and warning limit combinations across one or two data channels. The capability to solve for joint conditional relationships across one or more channels enables the method to potentially capture a wider range of conditional behaviour.

    - New parameter values are processed by the "core" modified SPC algorithm to produce three compound fitness metrics: the fitness score, the number of times the warning limits were exceeded, and the success rate. The metrics are suitable to allow comparisons across statistical features, ultimately enabling discovery of high-fitness statistical feature settings.

    - The intelligent agent can change settings describing how the genetic algorithms operate. Examples include the number of chromosomes in a population, the number of generations to use for training, the rates of mutation and cross-over, and a range of other settings. The agent can also modify the evaluation function settings in the search for improved method performance.

3. *Review combined statistical feature and limit results*. During and/or following training, the intelligent agent compares statistical feature settings using the test and validation data. Valuable statistical features capture most of the test data events with relatively few false alarms. Less valuable features capture fewer events and/or have relatively high levels of false alarms. The test data results provide feedback to the training process and training settings. When complete, an indication of the robustness of the settings can be achieved by analysing results obtained using the validation data.

4. *Generate composite event temporal pattern.* The intelligent agent places promising statistical feature and limit values into a neural network for composite event temporal recognition. The neural network attempts to find relationships, beyond those already identified, across statistical features from *x* data channels.

    - The intelligent agent controlling the neural network has the ability to evaluate the relative value of each input feature and reduce the number of features as necessary to produce the most robust results.

    - The intelligent agent can also change neural network settings to improve results.

    - Upon completion, an event prediction summary is generated describing the characteristics of each statistical feature and how each performed in the training/test/validation process.

5. *Monitor data stream(s).*

    - Monitor data stream(s) by reading new data and generating individual and composite event predictions.

    - Based on user preferences, if any warning limits on significant features are exceeded then notify the user. If no warning limits on individual features are exceeded but the composite temporal pattern indicates a warning is warranted, then notify the user.

Figure 3.1 reflects the information flow block diagram for developing the temporal event pattern, coinciding with Steps 1 through 4. A more complete discussion about these steps follows in Sections 3.2 through 3.5. Figure 3.2 depicts the information flow block diagram for monitoring data streams for the event pattern. Three activities are required to perform this function, introduced above as Step 5 and also discussed in Section 3.6.

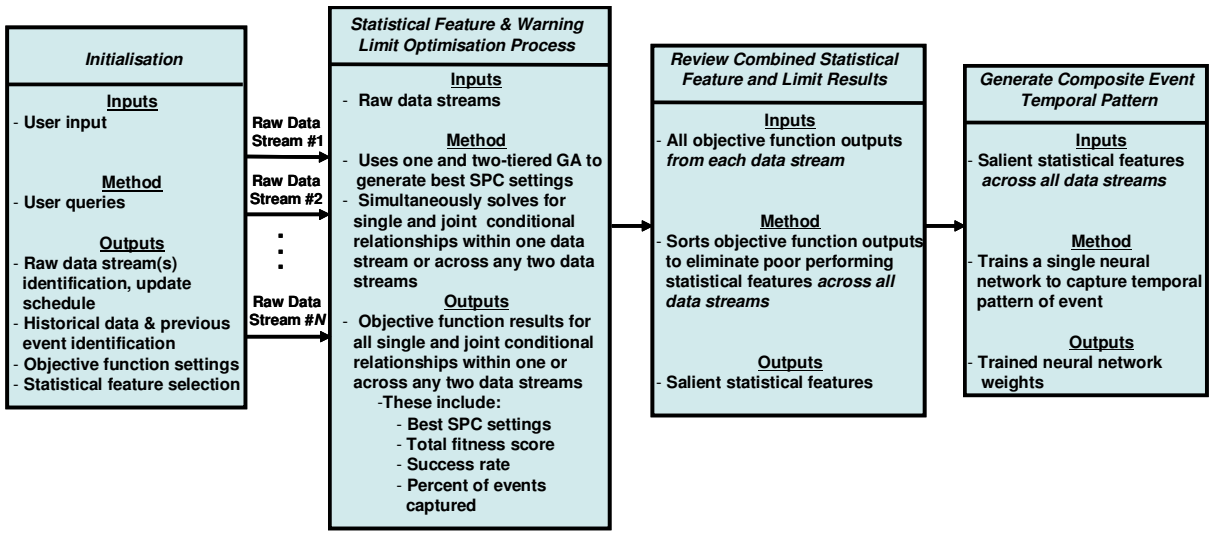

Figure 3.1. Block Flow Diagram for Developing the Temporal Event Pattern

3.2 Initialization

In the initialization task, the fundamental desires of the user are communicated to the intelligent agent. Many of the key elements were mentioned in Section 3.1. Table 3.1 details the range of objective

function choices available to the user. The method allows for separate weightings to be placed on two aspects of exceeding a limit: the weight given when a statistical feature simply exceeds a limit, and the weight given for the degree the limit was exceeded (the distance beyond the limit). Of particular interest is the option to explicitly identify the time window of acceptable alerts. Users can set when a pre-event alert is *too early* or *too late* to be useful. Furthermore, users can indicate their risk tolerance through the values they choose for true positive versus false positive alerts. Depending on the needs of the analysis and the perceived severity of each event, these point values may differ greatly. As shown in Table 3.1 and given mathematically in Equation 3.1, the objective function choices available to the user allow considerable flexibility for how the analysis is to be evaluated. Cumulative totals are tabulated by summing all points obtained over the training (or testing) data set as limits are exceeded.

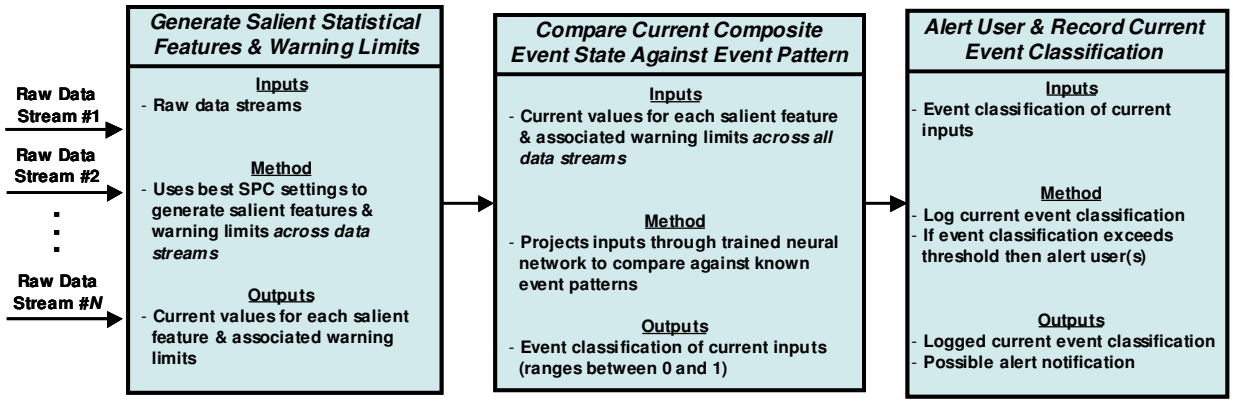

Figure 3.2. Block Flow Diagram for Monitoring Data Streams for Temporal Event Pattern

Correct Pre-Event Alarm Notifications	Values
Furthest # raw data points before event an alarm is deemed useful	10
Nearest # raw data points before event an alarm is deemed useful	0
Weighting on True Positive (TP) alarm notifications	1
Weighting on distance above/below limit for TP pre-event alarm notifications	1

Incorrect Pre-Event Alarm Notifications	Values
Weighting on False Positive (FP) pre-event alarm notifications	-1
Weighting on distance above/below limit for FP pre-event alarm notifications	1

During-Event Confirmatory Alarm Notifications	Values
Weighting on TP during-event alarm notifications	0
Weighting on distance above/below limit for TP during-event alarm notifications	1

Post-Event Confirmatory Alarm Notifications	Values
# raw data points after event completion an alarm is deemed confirmatory, neither TP or FP	10
Weighting on TP post-event alarm notifications	0
Weighting on distance above/below limit for TP post-event alarm notifications	1

Table 3.1. Objective Function Options

Understanding the range of different settings within Table 3.1 is easily accomplished. These particular settings indicate the user chooses to accept correct (true positive) pre-event alarm notifications within (and including) ten time periods of the actual event, giving each instance a weighting (or point value) of one. Each alarm notification during an event is given a weighting (or point value) of zero, as are post-event confirmatory alarm notifications up to ten time periods following an event. All other alarm notifications are deemed to be incorrect and result in the subtraction of one point, as evidenced by the negative weighting. The complete fitness score is calculated through the fitness function shown in Equation 3.1.

$$Score = (fw_b + \sum_{c=1}^{f} \alpha_j w_{kb}) + (gw_d + \sum_{c=1}^{g} \alpha_j w_{kd}) + (hw_a + \sum_{c=1}^{h} \alpha_j w_{ka}) - (lw_{FP} + \sum_{c=1}^{l} \alpha_j w_{FP}) \quad (3.1)$$

where

x_{eib} = the classification of a statistical observation exceeding one or more limits before the start of event e_i within time periods e_{iS}-b_1 and e_{iS}-b_2; considered a true - positive (TP) pre - event notification

x_{eid} = the classification of a statistical observation exceeding one or more limits between e_{iS} and e_{iT} of event e_i; considered a TP during - event notification

x_{eia} = the classification of a statistical observation exceeding one or more limits after e_{iT} of event e_i and before $e_{iT} + a$; considered a TP post - event (confirmatory) notification

x_{FP} = the classification of a statistical observation exceeding one or more limits and not classified as x_{eib}, x_{eid}, or x_{eia}; considered a false positive (FP).

e_{iS} = the start time of event e_i

e_{iT} = the stopping time of event e_i

b_1 = the earliest allowable pre - event notification of an event

b_2 = the latest allowable pre - event notification of an event

a = the maximum allowable number of post - event (confirmatory) time periods

x_j = the value of a statistical observation at time j

w_b = the weight assigned to an x_{eib} classification

w_d = the weight assigned to an x_{eid} classification

w_a = the weight assigned to an x_{eia} classification

w_{FP} = the weight assigned to an x_{FP} classification

w_{kb} = the weight assigned to α_j when classified as x_{eib}

w_{kd} = the weight assigned to α_j when classified as x_{eid}

w_{ka} = the weight assigned to α_j when classified as x_{eia}

w_{kFP} = the weight assigned to α_j when classified as x_{FP}

f = the count of all x_j classified as x_{eib}

g = the count of all x_j classified as x_{eid}

h = the count of all x_j classified as x_{eia}

l = the count of all x_j classified as x_{FP}

α_j = the absolute valued distance between observation x_j and the exceeded limit

3.3 Statistical feature and warning limit optimization process

The "core" of the method lies within the simultaneously solved statistical feature and warning limit optimization process. A single-layered genetic algorithm is used to solve for statistical features operating directly on the raw data, and a dual-layered genetic algorithm solves for statistical features operating on dependent data.

The method intentionally isolates the two primary elements within this process. The genetic algorithms generate parameter settings and receive fitness function results, whereas the "core"

modified SPC algorithm receives parameter settings and generates fitness function results. This separation increases code simplicity and enhances method generality. The process is discussed below, starting with the "core" modified SPC algorithm then followed by the genetic algorithm construct.

The modified SPC algorithm receives parameters from the genetic algorithm and processes them for fitness. To accomplish this task efficiently, the parameters are grouped by statistical feature type and associated with a data stream number. Figure 3.3 shows a fully completed sample configuration for Data Stream #2. Five features exist within the figure, broken down by "raw values", "directional trend", and "volatility". "Raw values" refers to processing the raw data using simple moving averages, "directional trend" refers to calculations involving velocity and acceleration, and "volatility" refers to calculations involving volatility and changes in volatility. These calculations are all very straight forward, and they can be performed over nearly any time window (depending on the statistical feature). The simple moving average is simply the rolling average of a given number of raw data values. Velocity is calculated as the rolling slope of a given number of raw data values, found by taking the slope following a first order least squares fit. Similar to velocity, acceleration is calculated as the rolling change in velocity of a given number of velocity data values, found by taking the slope following a first order least squares fit. Volatility is calculated as the rolling standard error of a first order least squares fit divided by the raw data average value within the time window. Similar to acceleration, change in volatility is calculated as the rolling change in volatility over a given time window, found by taking the slope following a first order least squares fit.

Event Detection Settings							
Data Stream #	2	Composite Score			Moodified SPC Settings		
					Initial Values	Current Values	Interim Best Settings
Raw Values	Simple Moving Average			Raw Data Length	11	11	11
	Metric	Summary Results		Trend Line MA Length	16	16	16
	Score	71		Trend Line StDev Length	377	377	377
	# Limits Exceeded	83		Upper Trend Line Limit k Value	1.6	1.6	1.6
	Success Rate	95.2%		Lower Trend Line Limit k Value	5.6	5.6	5.6
	Conduct GA Search?	0					
Directional Trend	Velocity (1st Moment)			Raw Data Length	20	20	20
	Metric	Summary Results		Trend Line MA Length	93	93	93
	Score	73		Trend Line StDev Length	1332	1332	1332
	# Limits Exceeded	80		Upper Trend Line Limit k Value	2.2	2.2	2.2
	Success Rate	97.5%		Lower Trend Line Limit k Value	3.8	3.8	3.8
	Conduct GA Search?	0					
	Acceleration (2nd Moment)			2nd Moment Raw Data Length	9	9	8
	Metric	Summary Results		Trend Line MA Length	909	909	904
	Score	17		Trend Line StDev Length	1210	1210	836
	# Limits Exceeded	53		Upper Trend Line Limit k Value	2.3	2.3	2.2
	Success Rate	84.9%		Lower Trend Line Limit k Value	6	6	8
	Conduct GA Search?	1		Source (Velocity) Raw Data Length	388	388	29
Volatility	Volatility (1st Moment)			Raw Data Length	29	29	29
	Metric	Summary Results		Trend Line MA Length	469	469	469
	Score	23		Trend Line StDev Length	1498	1498	1498
	# Limits Exceeded	26		Upper Trend Line Limit k Value	0.6	0.6	0.6
	Success Rate	100.0%		Lower Trend Line Limit k Value	7.8	7.8	7.8
	Conduct GA Search?	0					
	Change In Volatility (2nd Moment)			2nd Moment Raw Data Length	325	325	325
	Metric	Summary Results		Trend Line MA Length	262	262	262
	Score	-73		Trend Line StDev Length	274	274	274
	# Limits Exceeded	113		Upper Trend Line Limit k Value	2.5	2.5	2.5
	Success Rate	25.7%		Lower Trend Line Limit k Value	2.6	2.6	2.6
	Conduct GA Search?	0		Source (Volatility) Raw Data Length	319	319	319

Objective Function Settings	Values
Correct Pre-Event Alarm Notifications	
Furthest # raw data points before event an alarm is deemed useful	10
Nearest # raw data points before event an alarm is deemed useful	0
Weighting on True Positive (TP) alarm notifications	1
Weighting on distance above/below limit for TP pre-event alarm notifications	0
Incorrect Pre-Event Alarm Notifications	
Weighting on False Positive (FP) pre-event alarm notifications	-1
Weighting on distance above/below limit for FP pre-event alarm notifications	0
During-Event Confirmatory Alarm Notifications	
Weighting on TP during-event alarm notifications	0
Weighting on distance above/below limit for TP during-event alarm notifications	0
Post-Event Confirmatory Alarm Notifications	
# raw data points after event completion an alarm is deemed confirmatory, neither TP or FP	10
Weighting on TP post-event alarm notifications	0
Weighting on distance above/below limit for TP post-event alarm notifications	0

Figure 3.3. Sample Modified SPC Settings For Data Stream #2

For each statistical feature calculation, chart characteristics similar to those used within traditional SPC are identified. "Raw data length" refers to the number of raw data values within the rolling time window. In the case of velocity within Figure 3.3, the raw data length is 20, meaning 20 raw data values are included within the rolling velocity calculation. Each velocity calculation generates one instance, or *observation*, of the statistical feature. "Trend line MA length" refers to the number of observations contained within the trend line time window, a simple moving average (MA) of the observations. For velocity, the "trend line MA length" is 93, meaning the first velocity trend value is calculated starting with the 93rd velocity observation. "Upper trend line limit k value" refers to k in Equation 2.1, whereas "lower trend line limit k value" refers to k in Equation 2.2. "Trend line StDev Length" refers to n, also in Equations 2.1 and 2.2. Once the trend line moving average can be calculated, the respective limit lines are also calculated, even if n has not yet been reached. In this case the limit values will consist of n-m calculations, and m decreases to zero as the number of observations approach n.

Second moment calculations, like acceleration and change in volatility, require special processing by the algorithm. These calculations are dependent not on raw data directly, but on data produced following first moment calculations (velocity or volatility). Experimentation with the algorithm revealed optimal time window settings for first moment statistical features do not produce optimal time window settings for second moment statistical features. Accordingly, second moment calculations have an extra input line called "Source (velocity or volatility) Raw Data Length." This extra input provides the second moment with its own *direct link* to the raw data, and the second moment statistical features use the data from this calculation for it dependent statistical calculation. For instance, acceleration has a "source raw data length" of 388, meaning the input into the acceleration feature is velocity values calculated over a time window length of 388 raw data points. Following with this example, an acceleration value is generated upon receipt of the ninth velocity value. The additional layer of complexity associated with solving second moment statistical features necessitated developing a dual-layered genetic algorithm structure. One genetic algorithm labours to solve the "source raw data length" value, while the other genetic algorithm solves the remaining parameters for the statistical feature and its respective limits.

The method can be extended to solve more than one group of settings at a time, thus enabling joint conditional relationships to be considered and evaluated within one or across two data channels. The capability to identify significant joint conditional relationships within this step of the method offers the potential to vastly improve the method's final pattern recognition results. Adding a joint relationship capability to the method, however, requires modifying the evaluation routines to handle different types of relationships. New considerations include events defined by both statistical metrics simultaneously exceeding their limits or only one statistical metric exceeding its limits. The fitness function in Equation 3.1 still applies, however the evaluation routine needs must now consider each different type of relationship.

The method accommodates a wide range of different settings for each genetic algorithm layer to control the analysis process. Example settings are shown in Tables 3.2 and 3.3. Table 3.2 controls the first moment genetic algorithm and Table 3.3 provides controls for the second layer genetic algorithm. The genetic algorithms can perform "elite selection" where one copy of the best settings for a statistical feature (a single chromosome) is always carried forward into the next generation without interference from the crossover or mutation operators. Other settings for consideration include an option for automatically calculating the maximum time windows for the parameters, based on the proportion of the available raw data. Also shown in Table 3.3 is the option to change some of the settings for the primary genetic algorithm to reduce computational expense when solving second moment settings. The best set of settings is always made available to the user or intelligent agent should the training need to be stopped before normal stopping conditions are met. These are shown in Figure 3.3 under the column heading "Interim Best Settings."

The genetic algorithms optimize parameter settings through crossover and mutation operators and employ a roulette wheel selection process. The roulette wheel method is used for selection of parents from a population based on their fitness, shown in Figure 3.4. The outer ring is divided by the proportional contributions from each member of the population based on fitness. Random selection is used to determine which parent to select. Each genetic algorithm maintains a population of schema, representing groupings of parameter settings. With each generation, the genetic algorithms replace the current parent population with children schemata. In order for the selection method to operate, scaling of the fitness values is first accomplished as some fitness values fall below zero. The normalizing method described in [36] was used for this purpose. Once fitness values are normalized, chromosome selection can occur, and a schemata's probability of being selected is based on its proportional fitness. A certain percentage, p_c, are selected for crossover using the two cut point crossover approach. For those selected, a bit location is randomly chosen and the setting affected by the chosen bit exchanges its genetic material for the identical setting with the other selected chromosome. Finally, a small percentage of the child population, p_m, are randomly selected for mutation where a single bit is changed in value. Figure 3.5 and 3.6 show the crossover and mutation operations, respectively. With the mutation operation complete, the child population replaces the parent population, the fitness of the new population is evaluated, and the process repeats until a stopping condition is met.

Main Problem Variable Settings			
Variable Descriptions	Minimum Value	Maximum Value	Precision (# decimal places)
Raw Data Length	2	xxx	xxx
Trend Line MA Length	2	xxx	xxx
Trend Line StDev Length	3	xxx	xxx
Upper Trend Line Limit k Value	0	xxx	2
Lower Trend Line Limit k Value	0	xxx	2

Primary Genetic Algorithm Solver Settings	Value
Probability of crossover, pc	50.00%
Probability of mutation, pm	2.50%
Population Size (Number of chromosomes)	100
Max number of generations	500
Perform elite selection? (1 = Yes, 0 = No)	1

Automatically Set Maximum Setting Values? (1 = Yes, 0 = No)	1
Maximum Proportion Allocated To Raw Data Length	0.2
Maximum Proportion Allocated To Trend Line MA Length	0.5
Maximum Proportion Allocated To Trend Line StDev Length	0.9

Table 3.2. Max & Min Settings, Plus Primary Genetic Algorithm Control

Secondary "Source Raw Data" Problem Variable Settings		
Variable Descriptions	Minimum Value	Maximum Value
Data Length To Generate Raw Data For Acceleration & Change in Volatility	2	xxx

Secondary "Source Raw Data" Genetic Algorithm Solver Settings	Value
Probability of crossover, pc	0.6
Probability of mutation, pm	0.025
Population Size (Number of chromosomes)	40
Max number of generations	100
Perform elite selection (always maintain best known settings in next generation)? (1 = Yes, 0 = No)	1

Primary GA Settings During & After Running Secondary GA	Value
During Execution, Replace Primary GA with this Population Size (Number of chromosomes)	50
During Execution, Replace Primary GA with this Max number of generations	100
When Finished, Replace Primary GA with this Population Size (Number of chromosomes)	100
When Finished, Replace Primary GA with this Max number of generations	500

Automatically Set Maximum Setting Values? (1 = Yes, 0 = No)	1
Proportion Allocated To Raw Data Length	0.25

Table 3.3. Max & Min Settings, Plus Secondary Genetic Algorithm Control

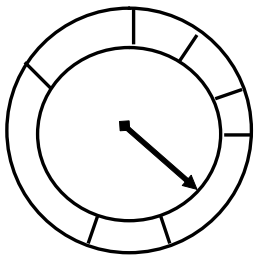

Figure 3.4. Roulette Wheel Selection

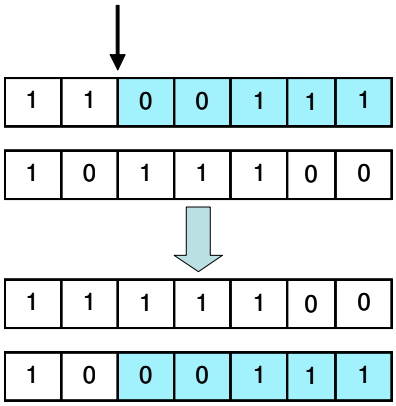

Figure 3.5. Crossover Operation

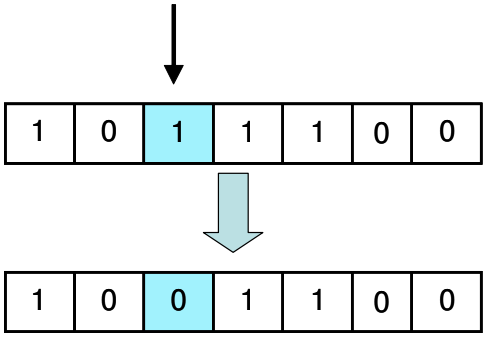

Figure 3.6. Mutation Operation

3.4 Review combined statistical feature and limit results

When training is completed, the summary metrics, combined with the number of events properly detected, can be used to evaluate how well each statistical feature performed. In general, the more consistently events are identified by features in the training data, the more likely they are to be found in the test and validation data. Features predicting a relatively small number of events in the training data are unlikely to be of further use, and the intelligent agent will not collect the data developed from those features.

3.5 Generate composite event temporal pattern

By combing statistical feature and limit combinations shown to be useful in the training data, the intelligent agent is able to develop a composite temporal pattern of the event. The composite data, consisting of one or more data streams, will then be evaluated by a neural network for overall event detection and prediction. The specifications (architecture and settings) for the neural network in this portion of the research have not yet been established.

3.6 Monitor data stream(s)

The method completes with an intelligent agent monitoring near real-time data from one or more data streams for event patterns. The monitoring activity handles data differently depending on the refresh rate of the incoming data and the user's settings for monitoring frequency. A low frequency monitoring rate, when more than several minutes pass between one monitoring update and the next, a "scheduled event" method is used to start and stop the activity. Each time the monitoring activity completes one update, the monitoring agent turns itself off after scheduling a future restart time on either the application or operating system level of the host computer. As each start of each monitoring activity requires parsing one or more historical data files and filling arrays to calculate the statistical metrics, computer resources are demanded periodically and then released. With low frequency

monitoring rates, these periodic demands are likely not consequential. Alternatively, high frequency monitoring rates, ranging from one second up to several minutes, requires immediate access to data arrays in order to complete all calculations prior to commencing the next monitoring cycle. Consequently, a special routine was developed to allow the host computer to perform all other operations while waiting for the next monitoring cycle to begin. This routine alternates between calling the "sleep" application programming interface (API) and the "DoEvents" command within Microsoft Visual Basic, as neither command by itself produces the desired result. By keeping the monitoring activity "live," recent historical data is only read once during monitoring function initialisation, high frequency monitoring can be achieved, and demands placed on the host computer are minimised.

When established limits are exceeded, the monitoring agent alerts one or more users to the potential of an (impending) event along with details about the alert including the statistical metrics involved and the observation and threshold values. Presently the method provides for two types of alerts: a pop-up message on the host computer and a broadcast e-mail alert. As users may or may not be present when a pop-up alert is generated, the monitoring activity continues to function in the background. All e-mail alerts are generated using Collaboration Data Objects (CDO) without accessing the host computer e-mail application. Recent advancements in anti-spam software have made auto-generated messages sent from computer to computer more difficult and secure networks frequently only allow authenticated message traffic. Depending on the network, inter-domain unauthenticated messages can still be generated and received. Cross-domain messages can be sent and received if the servers allow unauthenticated message traffic and/or blind message forwarding, conditions often met with most internet service providers. Adding authentication to the alerts is certainly possible and would overcome some of these difficulties.

Data can be made available to the data monitoring agent through passive or active means. Passive data availability implies an external mechanism retrieves data for access by the data monitoring agent. Active data collection is achieved through a data collection agent capable of retrieving data over the internet. The data collection agent aids the user in setting up the data collection process, collecting details such as the website addresses and key data fields on each site, as well as the data collection refresh rate. Despite the data collection agent's assistance, and depending on the complexity of the website layout and the type of data being collected, it is likely some computer programming is required to ensure the agent collects the correct data. Once completed, the user can select an option for continuous data collection at a certain frequency, or identify data collection over a range of dates through day/month/year criteria. Additional options are available including selecting individual days of the week, and event certain times within each day, for the collection to occur. At midnight each day the data collection agent determines and posts its data collection schedule for the next day. By posting its collection schedule and comparing this information to the last date and time data files are saved, an external "health" monitoring agent can determine if the data collection agent is operating properly or stopped prematurely due to unforeseen errors. A health monitoring agent, capable of monitoring the health of all the other agents through a similar fashion, has not been created but a need for its existence has already become apparent. The additional capability to monitor and automatically close and restart entire applications as needed offers the potential to create a very robust event detection and prediction system.

Chapter 4

Feasibility Study with Single Channelled Data

4.1 Overview

An initial feasibility study was conducted to evaluate the potential of the method and decide whether or not the broader outlined research objectives were reasonable and obtainable. This chapter presents the results of the study using single channelled data and a less mature version of the method. Chapter 5 presents similar material using two channelled real data and a more mature version of the method. Section 4.2 introduces the generation of two artificial data sets, and Section 4.3 discusses the analysis results using this data. Section 4.4 compares these results against other time series analysis methods.

4.2 Generating the Sample Data Sets and Adjusting Settings

To evaluate method feasibility, complex time series data needed to be acquired. A fundamental requirement was the need for significant control over how the data was generated. This section describes the generation process, and presents several graphs to enhance understanding.

The sample data streams each contained 5000 data points consisting of a complex signal sampled at 0.001 seconds. The complex signal contained four additive elements: two sinusoidal background signals, one partial sinusoidal event signal, and Gaussian noise. The first background signal was chosen to produce a non-stationary mean through an increasing constant positive slope over the data sets. As a result, it was a low frequency signal (0.01 Hz) with large amplitude of 50.0 and no phase shift. The second background signal had a higher frequency (1.0 Hz) with smaller amplitude of 1.0 and no phase shift. It was designed to add increased variability resulting in non-stationary variance within the data sets by randomly starting and stopping over the 5000 data points. Although random in duration, once the signal was randomly started it always completed at least two full periods. In addition, once the signal stopped, it always had to wait at least one period before starting again.

The event signal had a frequency of 20 Hz and no phase shift. It represented the signal to be identified by the method for correct event prediction. The sinusoidal signal started at $-\pi/2$ and continued until it reached $\pi/2$, resulting in a slanted s-shape appearance. The amplitude of the signal was either 1.0 or 1.5, depending on the data set, and the event was defined to occur at the peak value of the signal, $Asin(\pi/2)$, where A is the signal amplitude. There were no other differences between the two data sets besides the amplitude of the event signal. The event signal was randomly started and stopped throughout the 5000 data points within the time series, bounded by several constraints. First, events could not occur consecutively. Second, the number of time steps between events should average (over many thousands of data points) six periods. In this case, since there were 50 time steps per period in the event signal, over the long run the average number of time steps between events should be 300. A high variability factor was included in the random calculation to make the number of steps between events very irregular, as reflected in Table 4.1. It turns out the average number of time steps between events in the data streams was 401. The composite data stream was finalised by adding the signal values at each sampled time period. Gaussian noise, with a magnitude of 1.0, was then added to the composite data stream prior to analysis.

The method was provided the first 3000 data points within each data stream for training, with the remaining 2000 data points set aside for testing. Success or failure of the method is based on how well it performs on the testing data, since this represents data the method was unable to explore during the training process. Figures 4.1 and 4.2 show the completed data streams over 5000 data points, with a line separating the training and testing portions. A review of the data streams reveals several interesting points. One point to mention concerns how small the events appear when compared to the noise, a result of the low signal-to-noise ratio values. This is particularly true for the second data

stream where the signal and noise amplitudes are both 1.0. Another point to mention is the appearance of the higher frequency background signal only towards the end of the training data. This additional signal remains throughout roughly one third of the testing period. The significance of this observation concerns how little exposure the method has to this higher frequency within the training data, yet the overall feasibility of the method will be based on results obtained during the testing period where the signal is more prevalent.

Event Number	Time Of Event	Number of Data Points Between Events
1	27	---
2	507	480
3	1437	930
4	1859	422
5	2243	384
6	2454	211
7	2511	57
8	2624	113
9	2971	347
10	3309	338
11	3968	659
12	4273	305
13	4837	564

Table 4.1. Timing of Events and Number of Time Periods Between Events

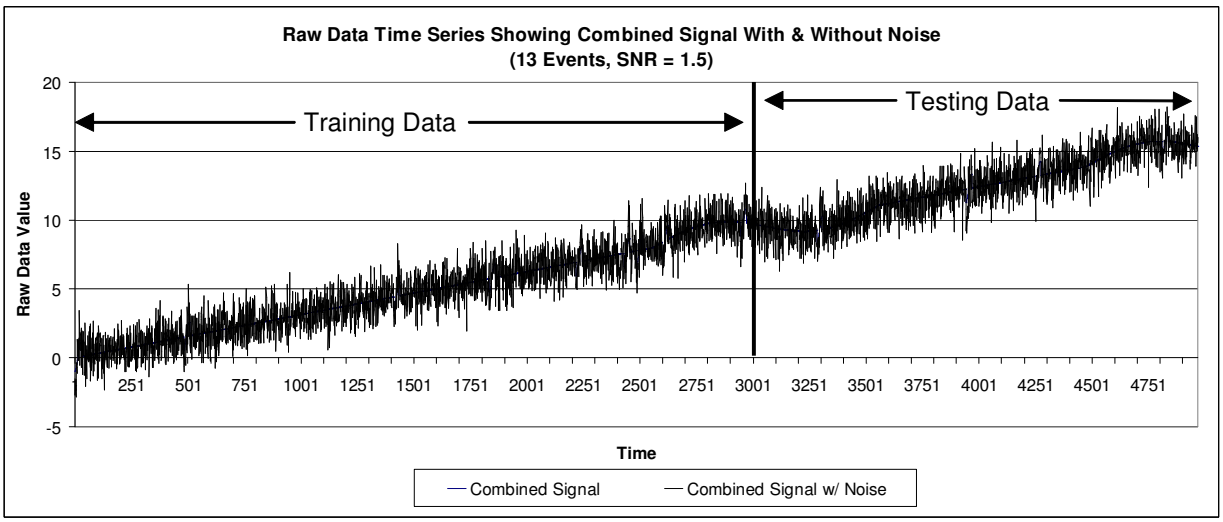

Figure 4.1. Complete Raw Time Series Data with SNR = 1.5

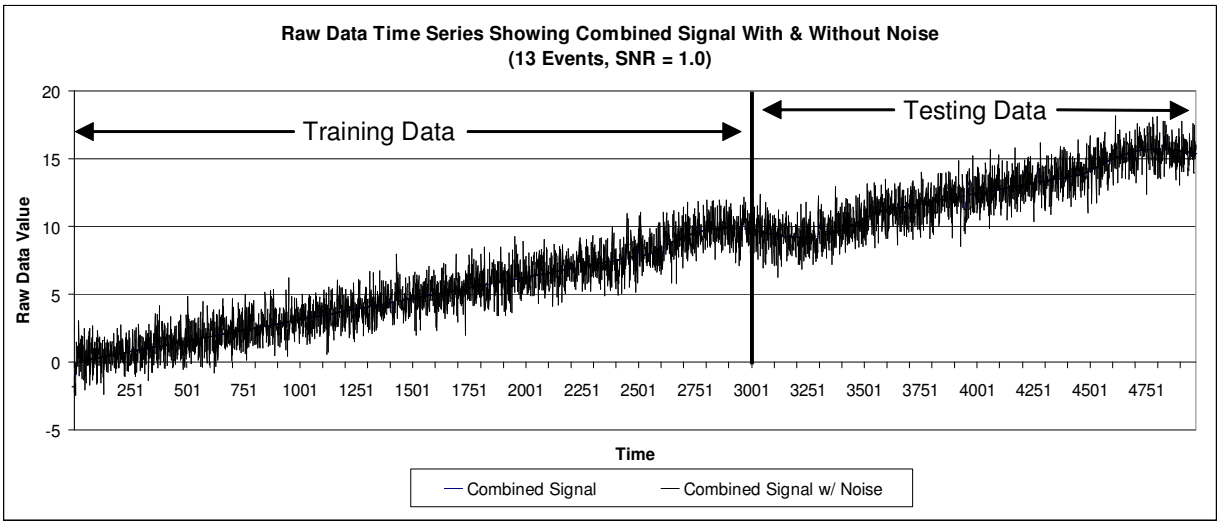

Figure 4.2. Complete Raw Time Series Data with SNR = 1.0

To make it easier to see how the events look within the data stream, two smaller sections of the data streams are presented in Figures 4.3 and 4.4. These figures each contain the same four events at times 2243, 2454, 2511, and 2624. They also more clearly show the composite data stream before noise was added, as well as showing the relative differences in event signal magnitude. The higher frequency background signal can be seen starting around time 2550.

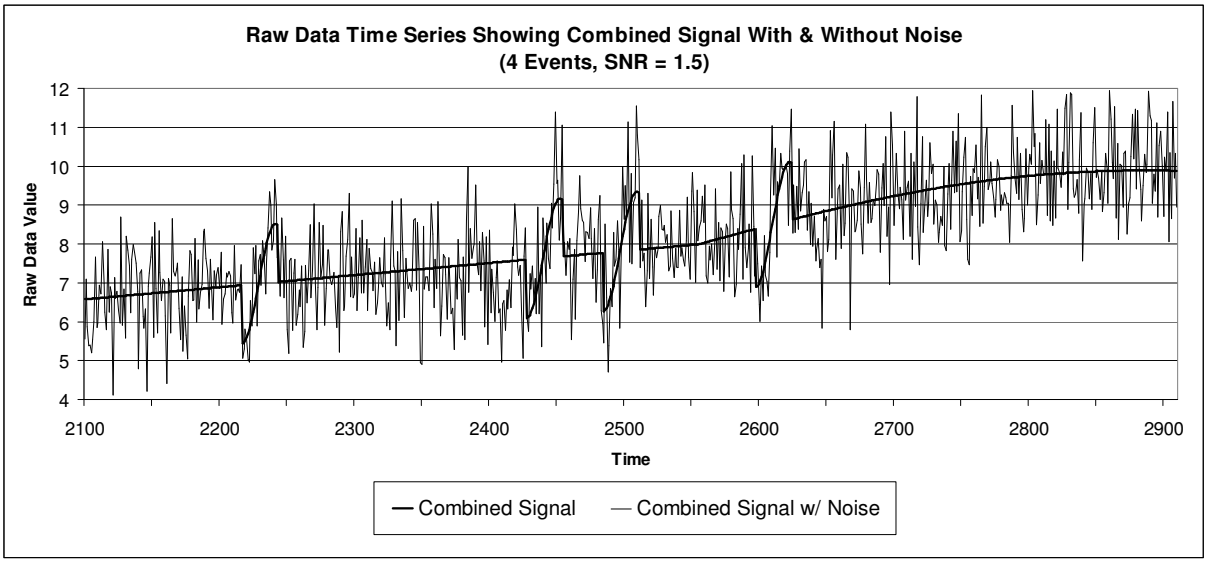

Figure 4.3. Magnified Section of Raw Time Series Data with SNR = 1.5

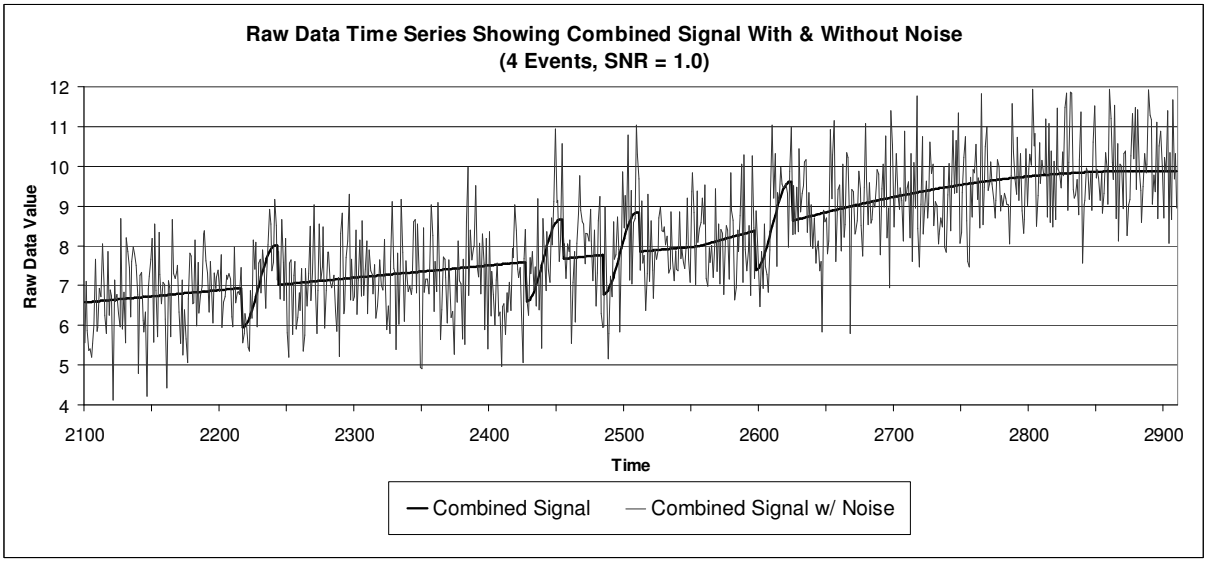

Figure 4.4. Magnified Section of Raw Time Series Data with SNR = 1.0

A spectral analysis of each data stream was performed. Figures 4.5 through 4.7 are the results of this analysis. A general overview of the frequencies contained within the raw data stream with SNR = 1.0 can be seen in Figure 4.5. The potential regions of interest are circled. The actual event signal, labelled in the figure, lies in a valley located at 20Hz. One also notices several larger spikes in the vicinity of the event signal, located at18.6Hz, 22.5Hz, and 26.9Hz. Several other spikes of nearly equal size are spread across a range of frequencies, with the highest frequency within an area of interest located at 240Hz. Figures 4.6 and 4.7 are magnifications of the low frequency spectrums to better highlight potentially interesting frequencies in the data. Analysis of the Figure 4.6 reveals generalized signal activity around the event frequency (20Hz). As noted earlier, the event frequency lies in a valley surrounded by two spikes. Figure 4.7 is almost identical to Figure 4.6, except the lower signal-to-noise ratio results in an overall reduction in power in the frequencies surrounding the event

signal. Given the low signal-to-noise ratios and randomness of the event signal within the non-stationary data, one would not expect remarkable insights from a spectral analysis. This is because frequency domain time series analysis requires stationary data (like most other time series analysis methods), an assumption clearly violated by the data. If suitable transformations were found to make the data stationary, then perhaps the spectral analysis would provide greater insight.

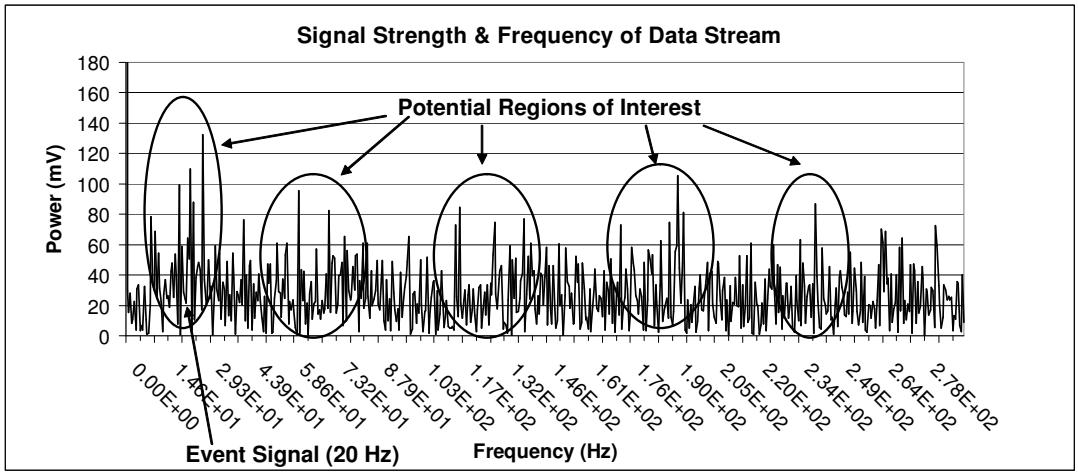

Figure 4.5. Overview of Frequencies Within Raw Data Stream with SNR = 1.0

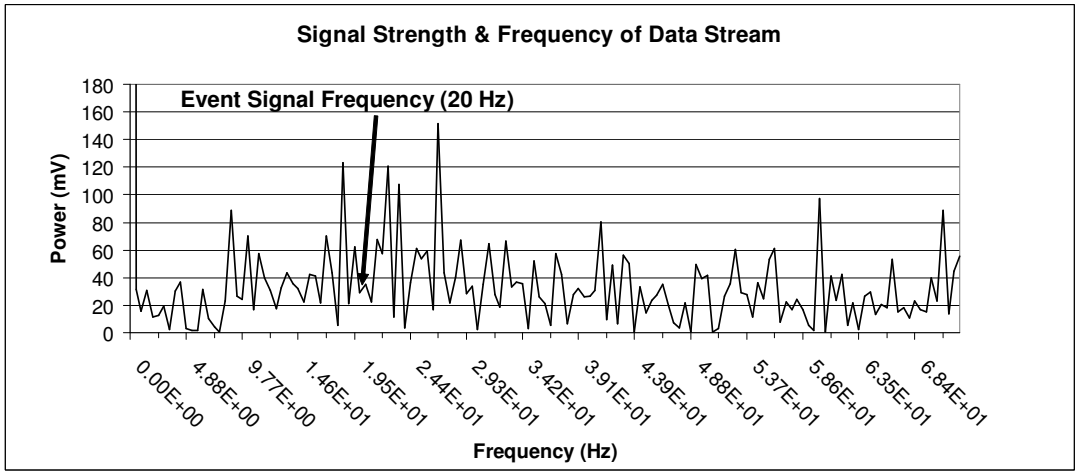

Figure 4.6. Magnification of the low frequency spectrum of Raw Data Stream with SNR = 1.5

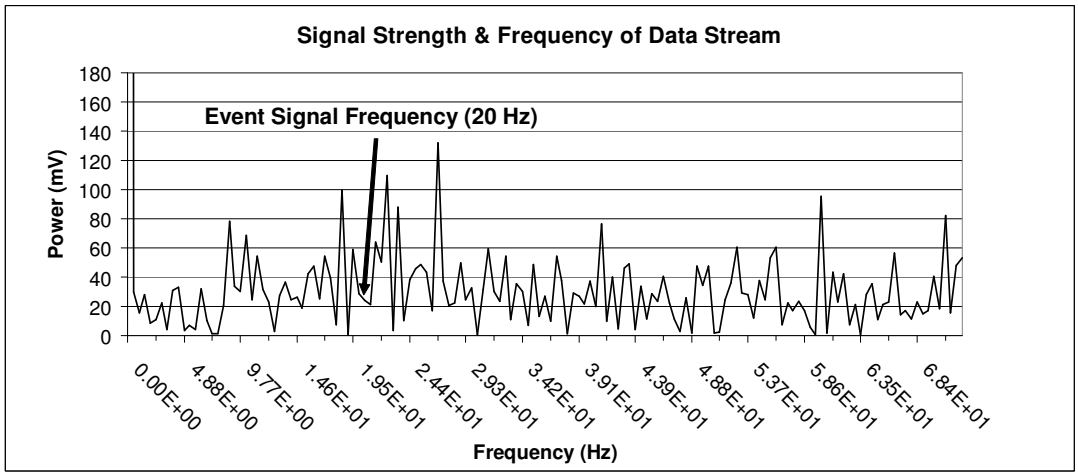

Figure 4.7. Magnification of the low frequency spectrum of Raw Data Stream with SNR = 1.0

The method requires two data elements in order to properly match the statistical features and warning limits to the events in the training data. One element is the raw training data, and the other element is the time when each event occurred. These elements were provided, and the objective function settings were entered as shown in Table 4.2, duplicated from Table 3.2 for convenience.

Correct Pre-Event Alarm Notifications	Values
Furthest # raw data points before event an alarm is deemed useful	10
Nearest # raw data points before event an alarm is deemed useful	0
Weighting on True Positive (TP) alarm notifications	1
Weighting on distance above/below limit for TP pre-event alarm notifications	0

Incorrect Pre-Event Alarm Notifications	Values
Weighting on False Positive (FP) pre-event alarm notifications	-1
Weighting on distance above/below limit for FP pre-event alarm notifications	0

During-Event Confirmatory Alarm Notifications	Values
Weighting on TP during-event alarm notifications	0
Weighting on distance above/below limit for TP during-event alarm notifications	0

Post-Event Confirmatory Alarm Notifications	Values
# raw data points after event completion an alarm is deemed confirmatory, neither TP or FP	10
Weighting on TP post-event alarm notifications	0
Weighting on distance above/below limit for TP post-event alarm notifications	0

Table 4.2. Objective Function Settings

While different combinations of settings could have been used, for feasibility purposes these seemed adequate. The settings reflect an equal weighting between correct and incorrect predictions (+1 or -1 points), and they allow up to 10 data points following an event for limits to be exceeded where points are neither awarded nor taken away. This allows for a short penalty-free "cool down" or "confirmatory" period should the statistical feature lie above (or below) a warning limit immediately following an event. Without allowing this concept and depending on the point values awarded (or taken away) for exceeded limits, the algorithm might find it optimal to never cross the warning limits, resulting in a useless search. Other settings included identifying the minimum and maximum values for each modified SPC setting, and the precision of the upper and lower trend limit lines. The minimum values were kept at the lowest possible setting, as shown in Table 4.3 (and duplicated from Table 3.3), and the maximum values were automatically calculated using the proportional settings shown at the bottom of Table 4.3. Other general settings to operate the primary genetic algorithm were as shown in the table. Table 4.4 provides the same basic information as Table 4.3, except the information controls the secondary layer genetic algorithm when solving dependent statistical features, such as acceleration and change in volatility. Also as shown in the table, users have the option to change some of the settings for the primary genetic algorithm to reduce computational expense. The values chosen are shown in the table.

Main Problem Variable Settings			
Variable Descriptions	Minimum Value	Maximum Value	Precision (# decimal places)
Raw Data Length	2	xxx	xxx
Trend Line MA Length	2	xxx	xxx
Trend Line StDev Length	3	xxx	xxx
Upper Trend Line Limit k Value	0	xxx	2
Lower Trend Line Limit k Value	0	xxx	2

Primary Genetic Algorithm Solver Settings	Value
Probability of crossover, pc	50.00%
Probability of mutation, pm	2.50%
Population Size (Number of chromosomes)	100
Max number of generations	500
Perform elite selection? (1 = Yes, 0 = No)	1

Automatically Set Maximum Setting Values? (1 = Yes, 0 = No)	1
Maximum Proportion Allocated To Raw Data Length	0.2
Maximum Proportion Allocated To Trend Line MA Length	0.5
Maximum Proportion Allocated To Trend Line StDev Length	0.9

Table 4.3. Max & Min Settings, Plus Primary Genetic Algorithm Control

Secondary "Source Raw Data" Problem Variable Settings		
Variable Descriptions	Minimum Value	Maximum Value
Data Length To Generate Raw Data For Acceleration & Change in Volatility	2	xxx

Secondary "Source Raw Data" Genetic Algorithm Solver Settings	Value
Probability of crossover, pc	0.6
Probability of mutation, pm	0.025
Population Size (Number of chromosomes)	40
Max number of generations	100
Perform elite selection (always maintain best known settings in next generation)? (1 = Yes, 0 = No)	1

Primary GA Settings During & After Running Secondary GA	Value
During Execution, Replace Primary GA with this Population Size (Number of chromosomes)	50
During Execution, Replace Primary GA with this Max number of generations	100
When Finished, Replace Primary GA with this Population Size (Number of chromosomes)	100
When Finished, Replace Primary GA with this Max number of generations	500

Automatically Set Maximum Setting Values? (1 = Yes, 0 = No)	1
Proportion Allocated To Raw Data Length	0.25

Table 4.4. Max & Min Settings, Plus Secondary Genetic Algorithm Control

4.3 Results of Analysis

As mentioned earlier, the way to identify whether or not the method is feasible should be based on how well the method performs during testing or validation periods following training over the training period. On the other hand, insight into how well each feature & limit combination may perform over the testing period can be provided by simply observing which feature & limit combinations correctly identified most of the events in the training period while maintaining a low false alarm rate. If a feature & limit combination detected a relatively small proportion of events, then it is unlikely to perform well in the testing period either.

In the case of the two data sets, two statistical feature and limit combinations performed well over both the training and testing periods. Those features were the simple moving average (MA) feature and the velocity feature. The simple moving average feature performed significantly better on the data set with the higher signal-to-noise ratio than with the lower ratio, whereas the velocity feature performed well on both data sets. The optimal settings found by the modified SPC method for both sets of features are located towards the end of the section in Table 4.9. Figures 4.8, 4.9, and 4.10 show graphically how the simple moving average feature and limit combination performed in the data set with SNR = 1.5. Figure 4.8 shows the combination across the entire time period, while Figures 4.9 and 4.10 show portions of data within the training and testing periods, respectively. In Figure 4.9, one will notice the simple moving average feature did not cross the warning limit line near time period 1960, thus avoiding a false positive. In Figure 4.10, one notices the feature correctly predicted events at time periods 3968 and 4273, but just missed the event at time period 4837. A summary of the results in tabular form is shown in Table 4.5.

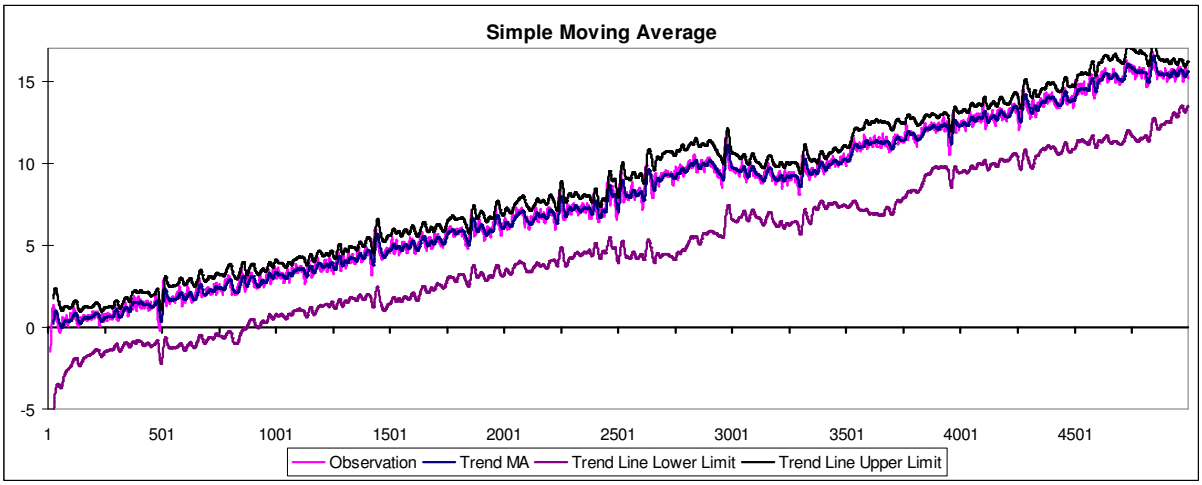

Figure 4.8. Results –Entire Simple Moving Average Feature & Limit Combination, SNR = 1.5

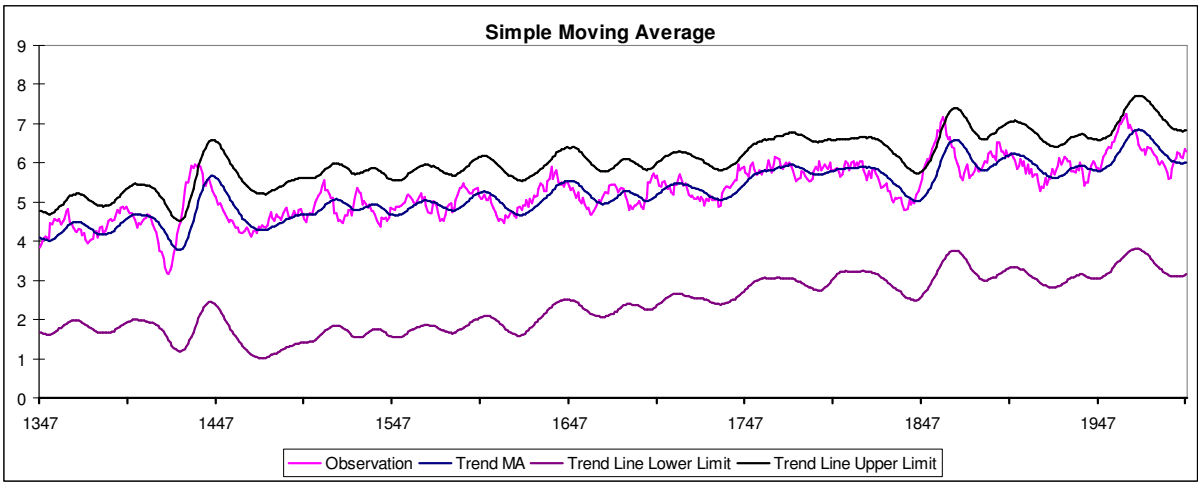

Figure 4.9. Results – Sample of Training Portion from Simple Moving Average Feature & Limit Combination, SNR = 1.5

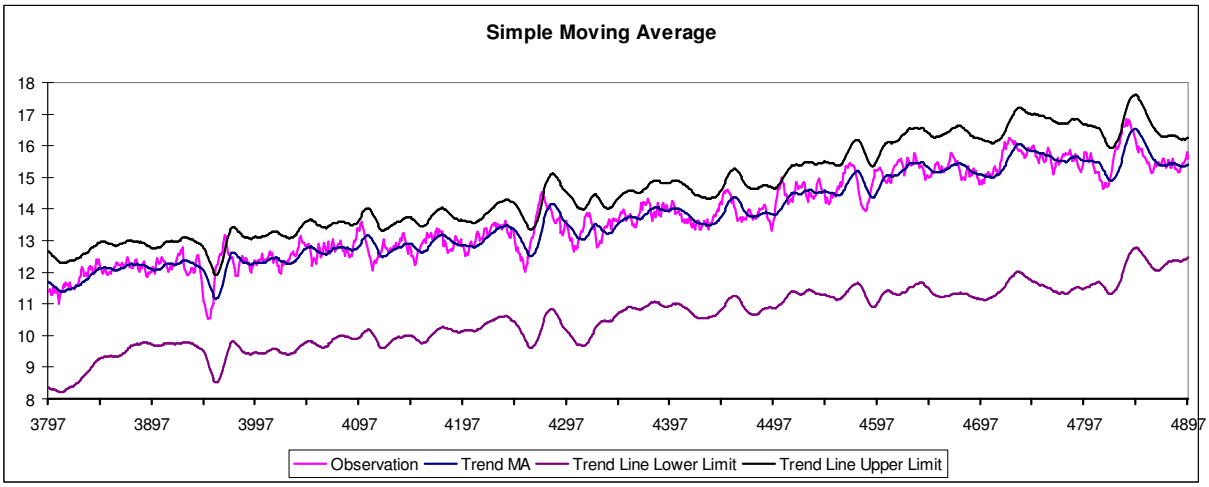

Figure 4.10. Results – Sample of Testing Portion from Simple Moving Average Feature & Limit Combination, SNR = 1.5

Number Time Periods Before Each Event An Alert Was Generated		
Time Of Event	**SNR = 1.0**	**SNR = 1.5**
27	**(limits not yet established)**	**(limits not yet established)**
507	**(limits not yet established)**	**9**
1437	**8**	**9**
1859	**1**	**10**
2243	**1**	**10**
2454	**4**	**6**
2511	**8**	**10**
2624	**8**	**8**
2971	**did not detect**	**8**
3309	**did not detect**	**8**
3968	**did not detect**	**9**
4273	**2**	**6**
4837	**did not detect**	**did not detect**
Average During Training Period	**5**	**8.75**
Average During Test Period	**2**	**7.67**

Table 4.5. Simple MA Results Summary – When Alerts Were Generated By Data Set

False Alarm Results		
Description	**SNR = 1.0**	**SNR = 1.5**
False Alarms In Training Period	**5 of out 3000 (0.17%)**	**4 of out 3000 (0.13%)**
False Alarms In Testing Period	**25 out of 2000 (1.25%)**	**2 out of 2000 (0.10%)**

Table 4.6. Simple MA Results Summary – False Alarm Statistics

Tables 4.5 and 4.6 show how many time periods before an event the feature produced a warning. In some cases this feature did not detect the event, particularly in the case of the data set with SNR = 1.0. The high average warning time in the data set with SNR = 1.5 also reflects the ease with which this feature could distinguish the event signal. In terms of false alarms, the feature produced very few. The most false alarms were recorded in the testing portion of the data set with the lower SNR, as expected.

While the simple moving average feature and limit combination produced solid results, even better results were obtained from the velocity feature and limit combination. Figures 4.11 and 4.12 graphically depict the results, while Tables 4.7 and 4.8 show results in a tabular format.

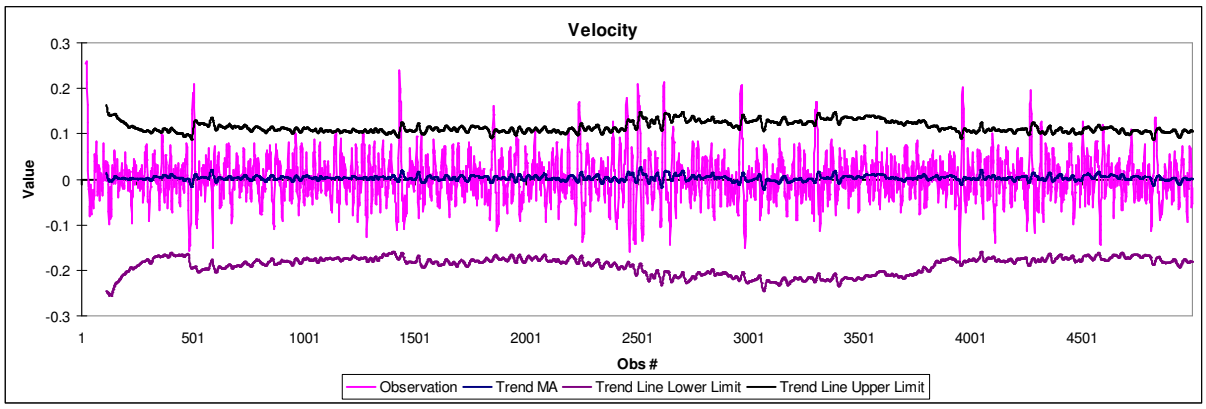

Figure 4.11. Results: Entire Velocity Feature & Limit Combination, SNR = 1.5

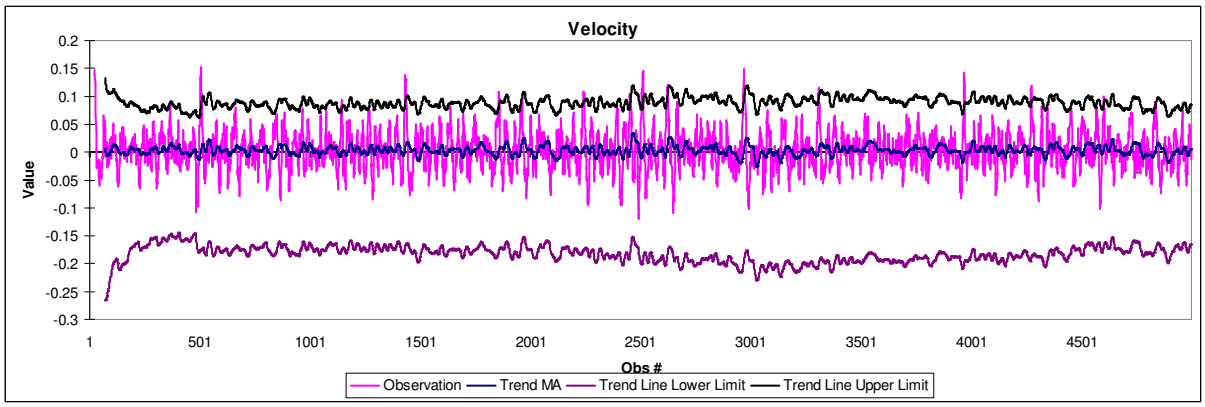

Figure 4.12. Results: Entire Velocity Feature & Limit Combination, SNR = 1.0

Figures 4.11 and 4.12 clearly show how well the statistical feature and limit combination performed in the two data sets. In both cases, the method was able to develop strong spikes prior to each event, allowing relatively easy event prediction to occur. In addition, the difference in event signal magnitudes between the two data sets is easy to distinguish. Glancing at Tables 4.7 and 4.8 also reveals how well this statistical feature and limit combination performed. Table 4.7 shows every event was predicted before it occurred, with a minimal warning time of three periods. It is interesting to note the average warning times only dropped slightly between the training and testing time periods, from 6 to 5, and from 8 to 7.25 for the two data sets. While an ideal case would reflect identical

values across the training and testing periods, the relatively small drop means the method produced a very robust classifier through the velocity feature and limit combination. It remains to be seen whether or not this holds true when other data sets, particularly using real data, are tested. Table 4.8 also portrays the strength of the velocity classifier for event detection by showing the false alarm rate stayed below 0.4%. The higher false alarm rate in the test data shown by the data set with SNR = 1.5 is caused by an attempt to have very early event detection. As with any warning system, the earlier the desired warning, the higher the expected false alarm rate [78].

Number Time Periods Before Each Event An Alert Was Generated		
Time Of Event	**SNR = 1.0**	**SNR = 1.5**
27	**(limits not yet established)**	**(limits not yet established)**
507	**7**	**9**
1437	**8**	**9**
1859	**4**	**8**
2243	**5**	**7**
2454	**4**	**6**
2511	**8**	**9**
2624	**5**	**8**
2971	**7**	**8**
3309	**4**	**7**
3968	**3**	**9**
4273	**4**	**6**
4837	**9**	**7**
Average During Training Period	**6**	**8**
Average During Test Period	**5**	**7.25**

Table 4.7. Velocity Results Summary – When Alerts Were Generated By Data Set

False Alarm Results		
Description	**SNR = 1.0**	**SNR = 1.5**
False Alarms In Training Period	**3 of out 3000 (0.10%)**	**0 of out 3000 (0.00%)**
False Alarms In Testing Period	**4 out of 2000 (0.20%)**	**7 out of 2000 (0.35%)**

Table 4.8. Velocity Results Summary – False Alarm Statistics

As mentioned earlier, Table 4.9 identifies the optimal settings developed from the method for both feature and limit combinations per data set. It is interesting to note that while some of the settings are remarkably close across data sets, some are very different. Larger gene pools and more generations might have reduced the differences observed within the table, however it is likely the change in signal amplitude relative to the noise is cause for most of the difference. The consistently wider limits on the data set with SNR = 1.0 confirms this notion. The largest difference is shown to be the trend line moving average length for the simple moving average. The trend line for the data set with SNR = 1.0 is so long that it practically resembles a straight line, whereas the trend line for the other data set is many times shorter. The stronger signal amplitude combined with the shorter trend line is likely one reason why this statistical feature was so much more successful in this example.

The calculations for this feasibility study were performed on a 3GHz hyperthreading computer with 1GB RAM running Windows XP Professional. In the right form, all of the statistical features can be calculated quickly. Research into exploring different features or variations of existing features yielded an improved form for the volatility equation involving no recursive operations, a significant advance over the initial volatility implementation. Outside the volatility metric, no other statistical features or changes to existing features were discovered. While the second moment calculations (acceleration and change in volatility) are individually fast, however, the dual-layered solution approach results in significant training time periods.

Optimal Settings By Statistical Feature Type Per Data Set			
Statistical Feature	Setting Description	Number of Time Periods Within Calculation	
		SNR = 1.0	SNR = 1.5
Simple Moving Average	Raw Data Length	9	11
	Trend Line MA Length	1405	16
	Trend Line StDev Length	661	377
	Upper Trend Line Limit k Value	4.33	1.6
	Lower Trend Line Limit k Value	6.99	5.6
Velocity	Raw Data Length	24	20
	Trend Line MA Length	48	93
	Trend Line StDev Length	1931	1332
	Upper Trend Line Limit k Value	2.42	2.2
	Lower Trend Line Limit k Value	5.27	3.8

Table 4.9. Optimal Settings For Both Feature and Limit Combinations Per Data Set

4.4 Comparison Against Other Time Series Analysis Methods

Several existing time series analysis methods were tested using the two data sets following multiple transformations to make them stationary. First, the data were differenced. This produced what appeared to be white noise, an independent variable for use in model building or placing in a neural network. Next a dependent variable was created to capture the timing of the events. The time of the event, plus the five previous time periods were all given values of one; all other time periods were given a value of zero. The raw and differenced data were then smoothed using three smoothing methods: exponentially smoothed, double (Brown) exponentially smoothed, and damped-trend linear exponentially smoothed. The total number of explanatory variables available for analysis was five. A linear model incorporating a full second-degree factorial of all the variables was attempted with no success (adjusted r^2 = 0.001229). Multiple combinations of these variables were also attempted, with no appreciable improvement. The best obtained adjusted r^2 value was 0.06743. Multiple autoregressive integrated moving average (ARIMA) models were then build to see if the raw data, differenced data, or smoothed data could be improved to the point of helping the linear modelling effort (still applying a full second degree factorial of all the variables). Despite lots of effort and roughly seventy ARIMA combinations (including seasonal adjustments), the effort was stopped after having only reached slight improvement. The best adjusted r^2 value was found to be 0.1814. Successful results were not expected from linear or ARIMA-based modelling techniques, and the results confirmed these expectations.

Building a non-linear model requires knowing the functional form of the data, a condition unlikely to be met in without substantial *apriori* knowledge of the underlying phenomenon generating the time-series, so this option was discarded. Fifteen pattern recognition neural networks were then evaluated using the variables listed above (the raw data, differenced data, and smoothed data) against the independent event variable. Different numbers of hidden nodes were chosen (5, 10, and 15), and two learning rates were tried (0.02 and 0.1). The momentum variable was held constant at 0.20, and training was terminated after 100 epochs (if not already terminated due to increasing hold-out sample root mean square error values). Unfortunately, none of the neural networks performed any better than the linear models on this data set. The best one predicted one event and correctly classified three out of 20 pre-event time periods. The neural networks essentially always misclassified the events as non-events, a common result when only a few data points are classified differently than the majority.

More time could have been spent trying other time series analysis methods, but it became clear that little improvement would be found without lots of trial-and-error. The extended time and frustration in attempting to analyze the complex data using the various methods described above underscores why an easy-to-use, more robust method needs to be developed.

Chapter 5

Feasibility Study with Two Channelled Real Data

5.1 Overview

A feasibility study using two channelled data from a real environmental process was conducted to further evaluate the potential of the method. This chapter presents the results of the study using river data collected from Pajaro River in Chittenden, California, USA. The method employed for this study uses a more advanced version of the first stage pattern recognition method than found in Chapter 4. Differences between the two versions include the capability to analyse joint conditional relationships, the introduction of constant-valued threshold limits in addition to trend-based limits, and the capability to create event "bands" by allowing the lower and upper limits complete freedom in their selection of k values. Section 5.2 introduces the data used in the analysis, Section 5.3 discusses how the data was prepared for analysis, Section 5.4 shows results for characterising a flood event, and Section 5.5 shows results for predicting a flood event.

5.2 Introduction to the River Data

The data acquired for this feasibility study came from the United States Geological Survey (USGS) and covers the Pajaro River near Chittenden located at 36.9020°N latitude and 121.6050°W longitude. The testing site collects a range of river-related data and is stationed at an elevation of 82ft above sea level. The data used here was collected over 20 years, from 1984 through 2004, and included hourly measurements for water flow, measured in cubic feet per second (cfs), and water stage, measured in feet (ft).

The USGS defines a flood for this river at this location as any time the water stage meets or exceeds 32ft in height. A "monitor" stage, or flood warning, occurs whenever the water stage meets or exceeds 25ft. Over the 20-year monitoring period, two unique floods occur. The first flood, detailed in Figures 5.1 and 5.2, occurred on March 10th in 1995. Flood stage conditions only lasted for seven hours over an 18-hour period. As shown in Figure 5.1, the water only slightly exceeded the flood stage during the course of the flood period, and the data shows a maximum recorded water stage of 32.13ft. The 32ft level was exceeded for four hours, dropping below 32ft for the next 12 hours, and then exceeding 32ft again for three more hours before slowly receding back to more normal levels. Importantly, there were a total of four hours between the flood warning and the flood, the duration of time when the water exceeded the "monitor" stage prior to exceeding the flood stage. The flood is recorded to occur on the fifth hour following the flood warning. According to the USGS this was a 35-year flood (meaning a flood of this magnitude is expected to occur about once every 35 years). The second flood, occurring between February 2nd and 3rd, 1998, and shown in Figures 5.3 and 5.4, exceeded the flood stage for seventeen straight hours and reached a maximum recorded water stage of 34.92 feet. The 25ft flood warning stage was exceeded for eleven hours prior to the reaching the flood stage on the 12th hour. According to the USGS, a flood of this magnitude is expected to occur about once about every 80 years.

As shown in Figures 5.1 through 5.4, both data channels exhibit similar overall characteristics during a flood event as water flow and water stage increase. Different rates of increase and decrease are also noted, with sharper increases and decreases found in the water flow data. The entire training and validation data sets used within this analysis are shown in Figures 5.5 through 5.8. Visible plateaus in the data sets are due to pockets of bad or missing data and not due to natural environmental causes, as discussed more fully in Section 5.3.

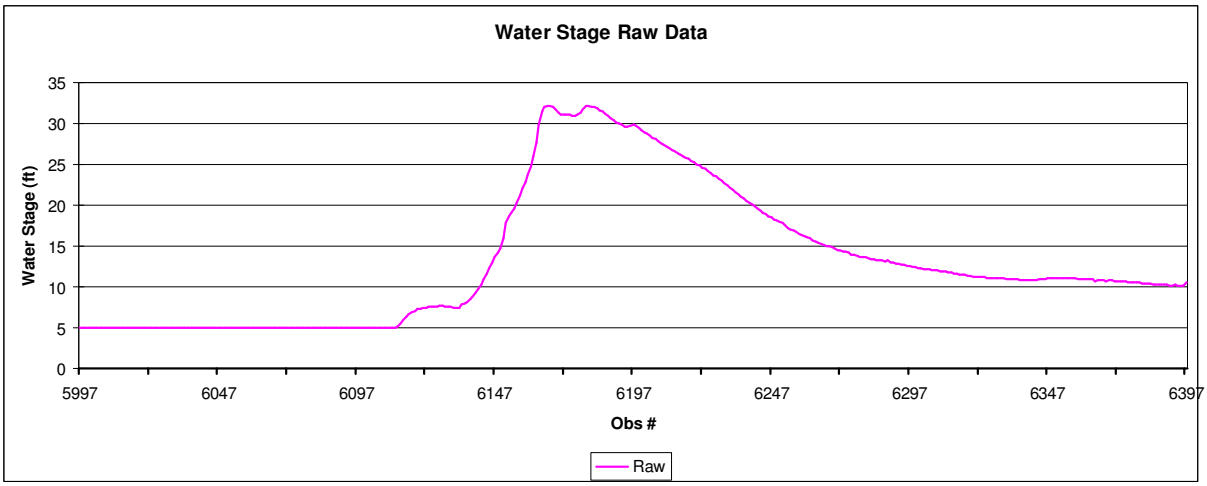

Figure 5.1. Review of March 1995 Flood: Water Stage Data

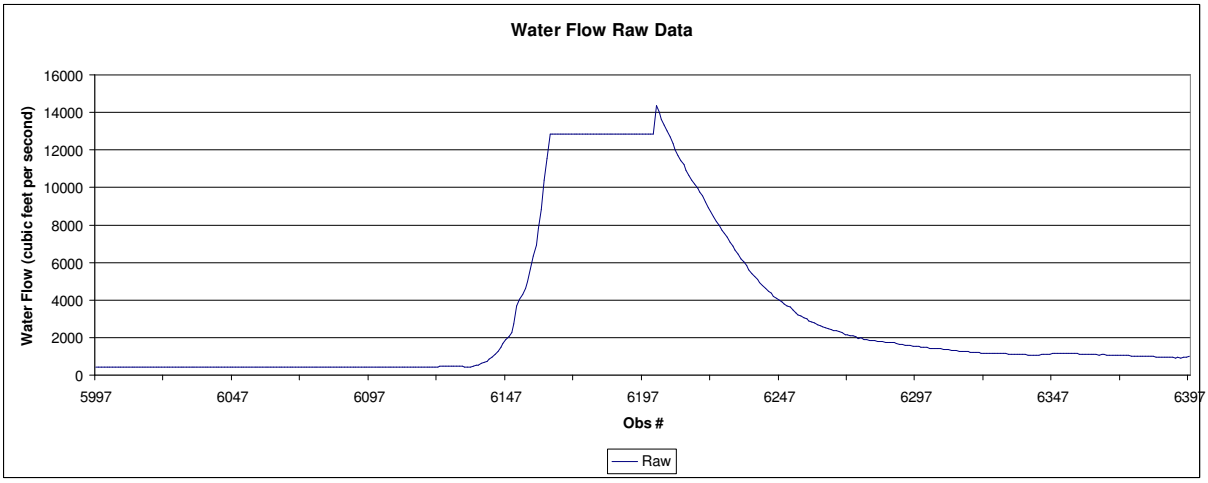

Figure 5.2. Review of March 1995 Flood: Water Flow Data

5.3 Pre-processing the River Data and Generating Data Sets

The USGS maintains thousands of data collection sites throughout the United States monitoring a wide range of environmental phenomena. It is not surprising, therefore, that data are not always recorded as desired due to any number of reasons. The two channelled data studied here had to undergo significant pre-processing to enable its use, and the method to accomplish this pre-processing is presented below. The overall pre-processing effort involved multiple passes through the data containing over 183,000 data points each. Each pass performed different functions such as identifying and handling obvious outliers and erroneous data, and filling in missing values where gaps were discovered.

The first step entailed finding the median of each data set, a metric of central tendency less susceptible to outliers than the mean. Extreme deviations from one data point to the next, more than two orders of magnitude, were ignored in the median calculation as were all negative and non-numerical values. The resulting median values were low, coinciding with California's long standing water shortage concerns. To allow the median values to be useful in identifying less obvious outliers, minimum acceptable data set values were established by multiplying the median values by 2.5. The second step entailed upwardly adjusting all values falling below the minimum values. The third step attempted to identify less obvious outliers through a look-forward, look-backward process. With each time step, neighbouring values exceeding 300% of the preceding value were reduced to the average between the following two data points and the previous two data points. In the fourth step, the data was reviewed

to identify and fill missing data points, taking into account the number of days in each month and even leap year considerations. Since many of the missing time gaps were large, ranging from several hours to several months, the gaps were filled with the preceding data point value. The effect of introducing a minimum value and filling missing data in this manner generated plateaus of short or long duration.

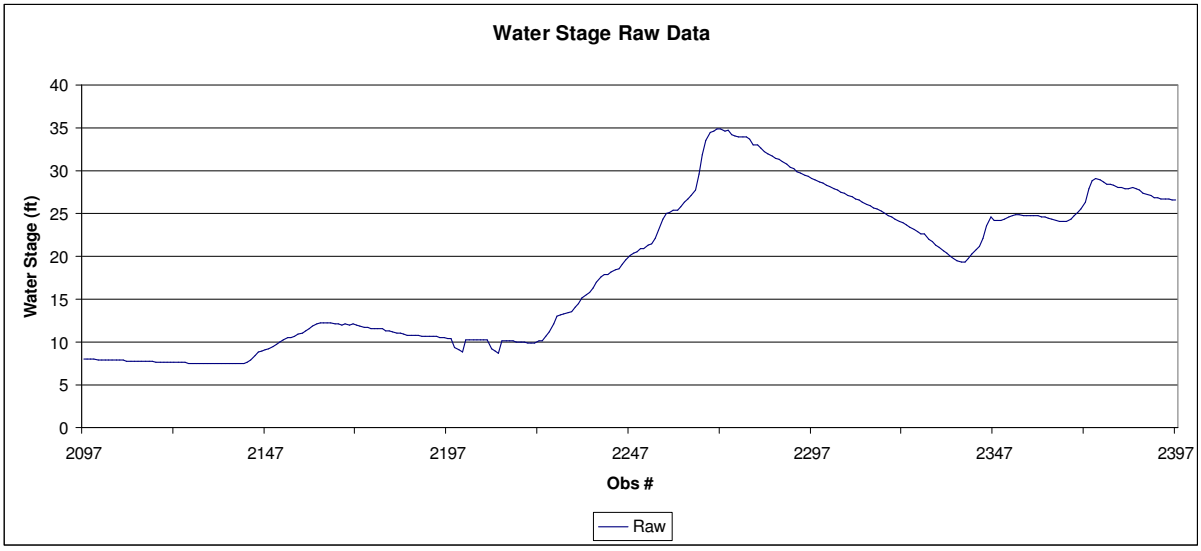

Figure 5.3. Review of February 1998 Flood: Water Stage Data

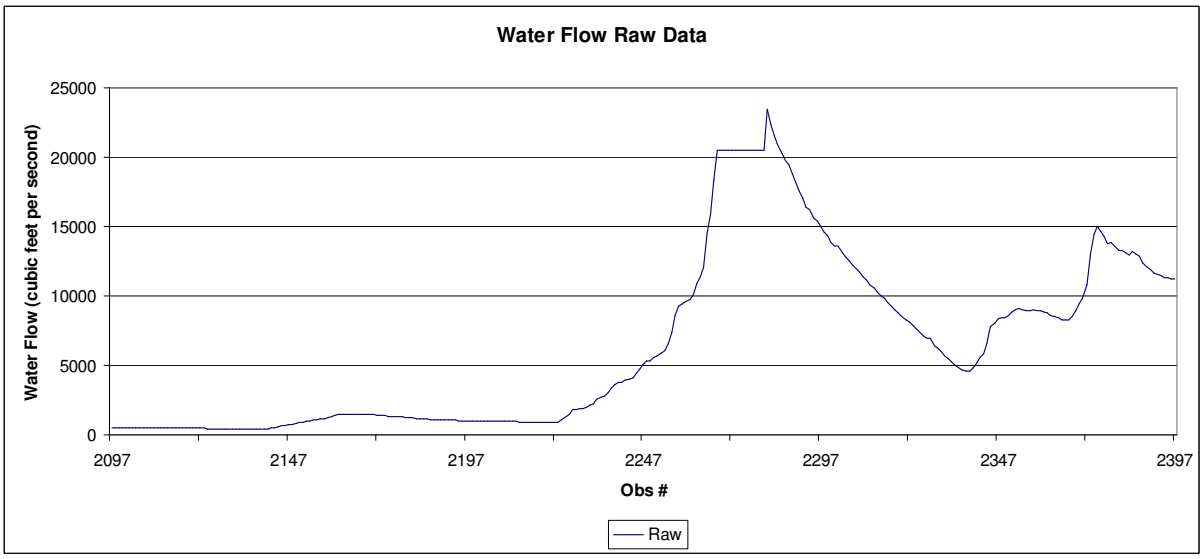

Figure 5.4. Review of February 1998 Flood: Water Flow Data

By far the most frequent plateau was the contrived minimum value, as the Pajaro River experienced extended periods of low water flow and low water stage levels. Consequently, useful data sets were constructed by collecting each significant increase in the data streams and piecing them together to generate new data sets, leaving several hundred data points on each side of the increases intact. The metric used to determine a significant increase was defined as any period where the water stage increased beyond ten feet, a value less than one third of the flood stage. This process resulted in two reduced data sets each containing 30,095 data points spanning the years between 1984 and 2004. As mentioned earlier, over this timeframe there were two floods and approximately thirty-three instances where the water stage increased beyond ten feet.

Ideally, three data sets would be constructed using the available data: a training set, a testing set, and a validation set. Each data set would contain multiple examples of events to ensure the most robust settings were established during the training process. The most robust settings are typically identified as those producing the most favourable results on the test data set, and an indication of the robustness can be achieved by analysing results obtained using the validation data set. Using less than three data sets increases the risk of over-fitting the data, resulting in settings unable to properly identify events in variable complex processes.

With only two flood events in the entire data set to use for reference, the decision was made to develop only two data sets for each data channel: a training set and a validation set. The results of the pre-processing effort are shown in Figures 5.5 through 5.8.

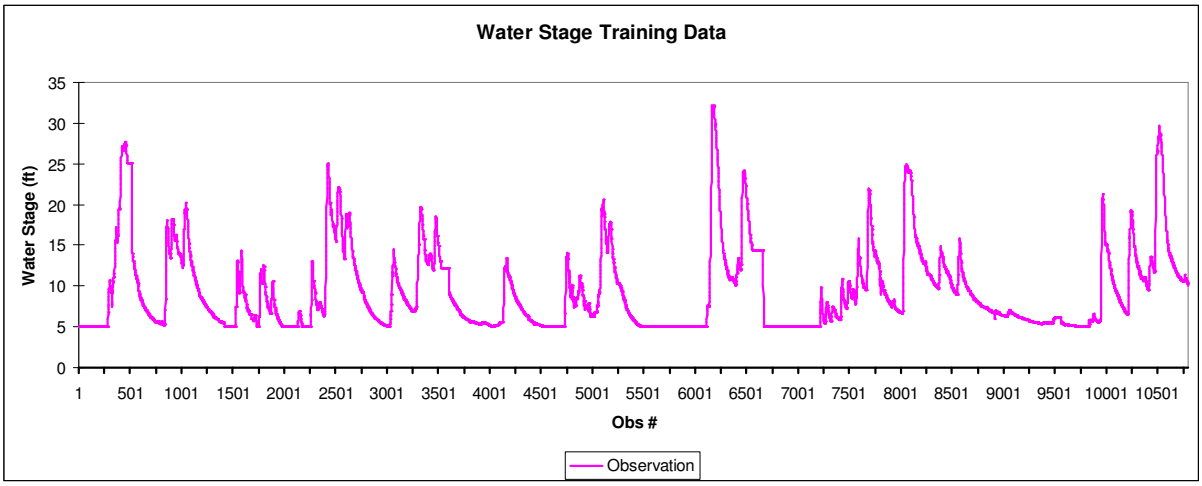

Figure 5.5. Review of Training Data Set: Water Stage

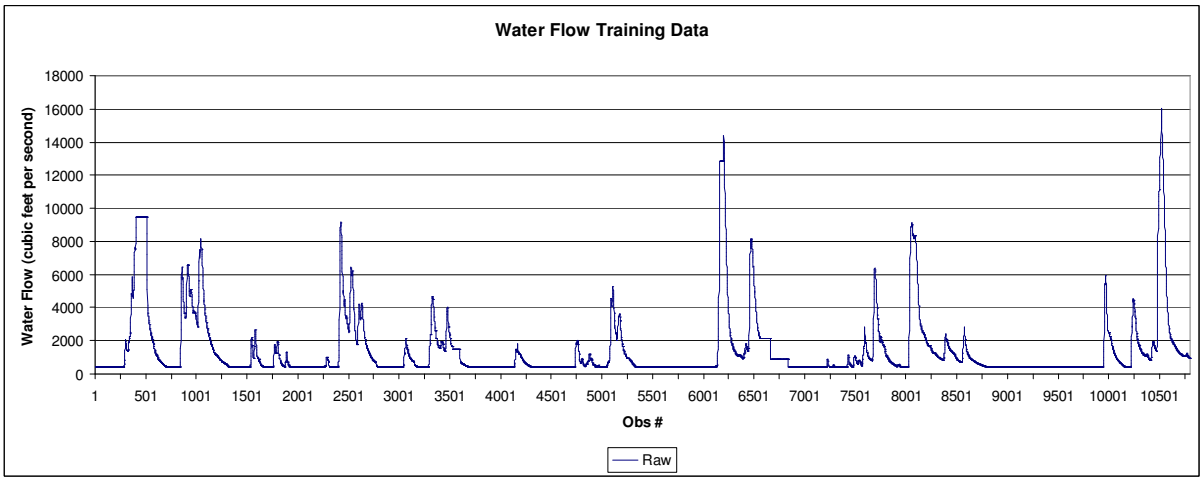

Figure 5.6. Review of Training Data Set: Water Flow

5.4 Characterising a Flood Event

A few fundamental questions arise when working with event-related temporal data. One question considers whether or not the method can characterise the event of interest, and another considers whether or not the method can predict the event. This section discusses an attempt to characterise the flood event.

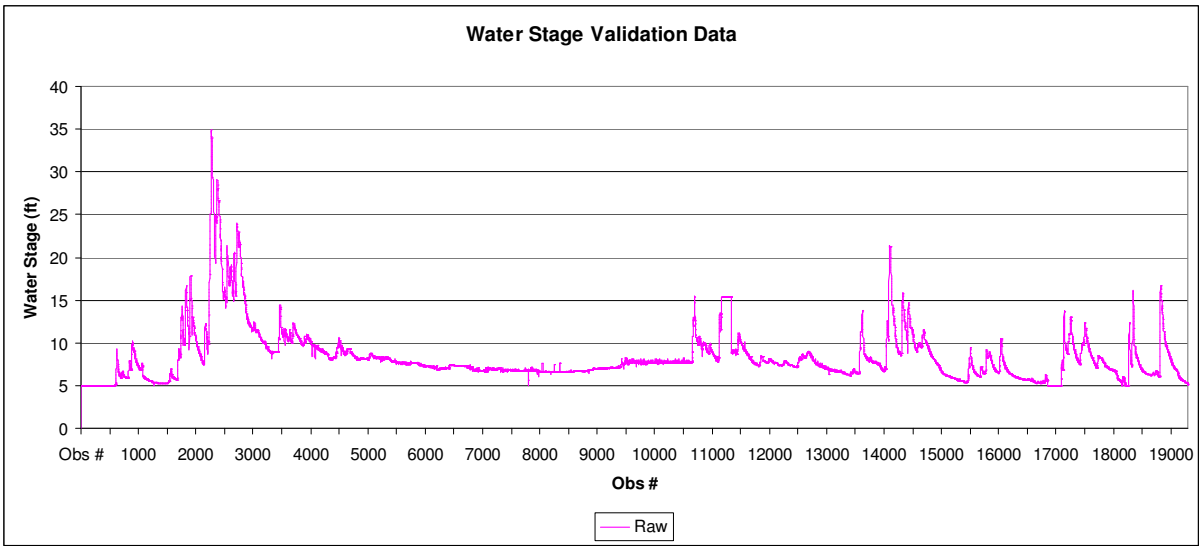

Figure 5.7. Review of Validation Data Set: Water Stage

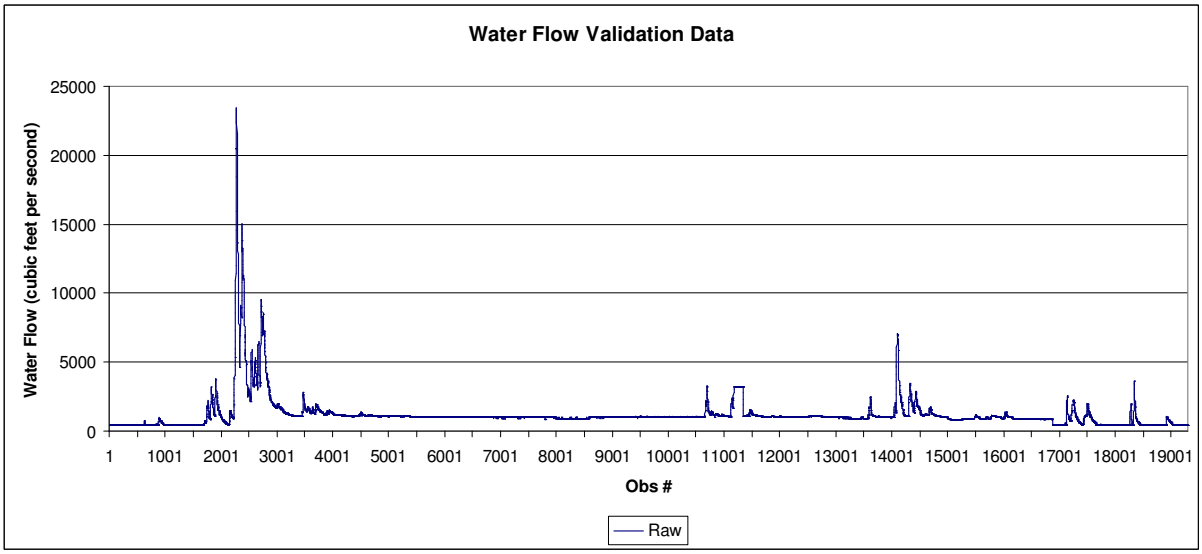

Figure 5.8. Review of Validation Data Set: Water Flow

First and foremost, however, it must be *clearly* understood the results presented in Chapter 5 are unfortunately based on a poor selection of data sets where only two flood events occurred during the monitored time period. There is no doubt training the method on only one event will result in over-fitting that event, thus severely limiting the robustness of the resultant settings. This problem is due to the data and *not* the method. As a result, comprehensive finalised settings will not be presented in this chapter and significant conclusions cannot be reached.

An attempt was made to limit the effects of over-fitting the training data. Since genetic algorithms refine possible solutions with each generation of chromosomes, one way to combat the over-fitting problem involves reducing the number of generations before the best settings are finalised. This approach was chosen and the number of generations was reduced from 200 down to 20, however no sensitivity analysis was performed to determine whether or not 20 was an appropriate value. Other genetic algorithm settings included a crossover probability of 0.65, a mutation probability of 0.035, and a population size of 200. Elite selection was not employed. The following objective function weights were used to characterise, *not predict*, the flood event: +1 for each TP during-event notification, w_b, and -1 for all other notifications. A value of zero was given to α_i.

The results, shown in Figures 5.9 and 5.10, indicate the method was easily able to characterise *this* event. It achieved perfect success using the water stage data by combining a simple moving average over one time step in conjunction with a constant-valued limit placed at 31.96ft. (In this example, the method generated an event "band" between 31.96ft, the upper limit, and 35.92ft, the lower limit.) Since 32ft is defined as a flood, the value obtained by the genetic algorithm is completely accurate. While promising, more excitement would be generated with this result had more events existed within the training data.

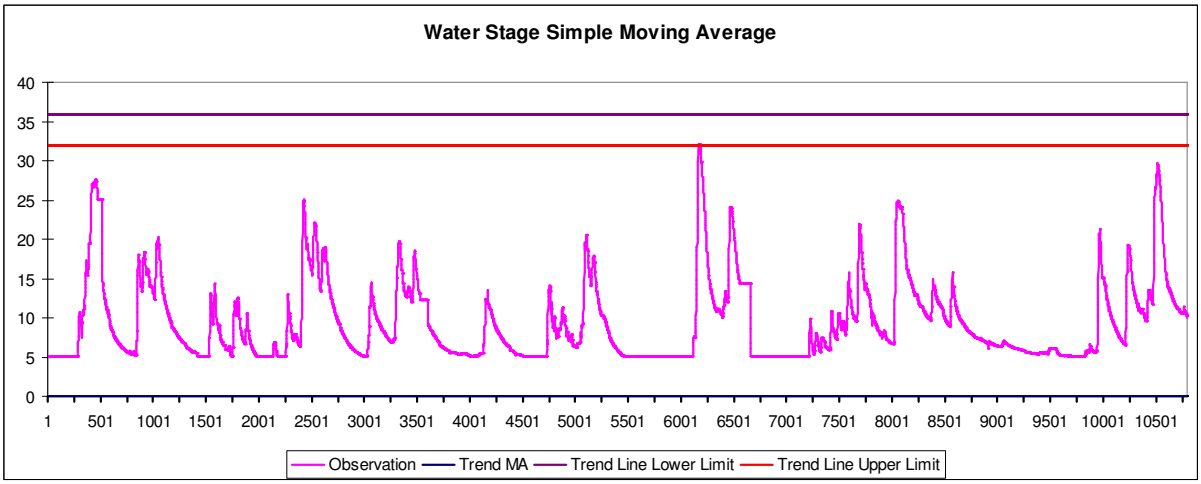

Figure 5.9. Overview of Flood Event Characterisation

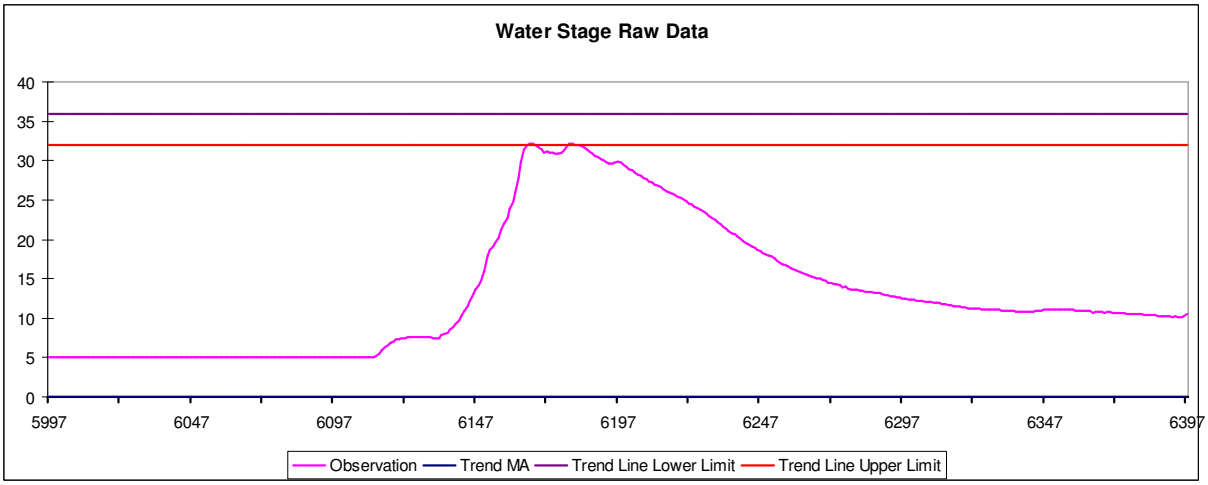

Figure 5.10. Detailed Review of Flood Event Characterisation

5.5 Predicting a Flood Event

Only three settings needed to be changed when transitioning from trying to characterise the event to trying to predict the event: b_1 (changed to 20), a (changed to 20), and w_b (changed to +1).

All combinations of statistical features across both data streams were evaluated in the course of the analysis. The results, shown in summary format in Figure 5.11 and in detail in the Appendix, indicate wide performance differences across the different statistical features. Figure 5.11 only shows those features found to have successfully produced one or more pre-event notifications in the training data set. For each feature, an average has been computed reflecting how it performed by itself and jointly

in the validation data set. The average metric reflects the average number of pre-event notification periods in the *validation* data set, and the number of attempts identifies how many combinations with that metric detected the flood event in the *validation* data set. For example, the best overall results came from water flow velocity. This statistical feature not only performed well in the training data, as did many others, but also produced an average pre-event notification of 2.4 time periods before the flood event in the validation data set. The 2.4 average is based over seven combinations with other statistical metrics, indicating water flow velocity is likely a very significant feature for flood detection and prediction. The next best performing statistical feature appears to be the simple moving average of the water stage data. After a more detailed review of the data, however, one discovers it actually only had a very good result when paired with itself and only moderate results when paired with the other statistical features. When paired with itself, two simple moving averages with different settings over different time windows were created, achieving ten periods of pre-event notification for the flood event in the validation data. Solid results also appear in the water flow simple moving average statistical metric, however Figure 5.11 indicates the results were not as consistent across as many statistical metrics as the water flow velocity and water stage simple moving average statistical metrics. As mentioned earlier, however, while these results are certainly interesting no significant conclusions can be drawn from them due to the low event count in the data.

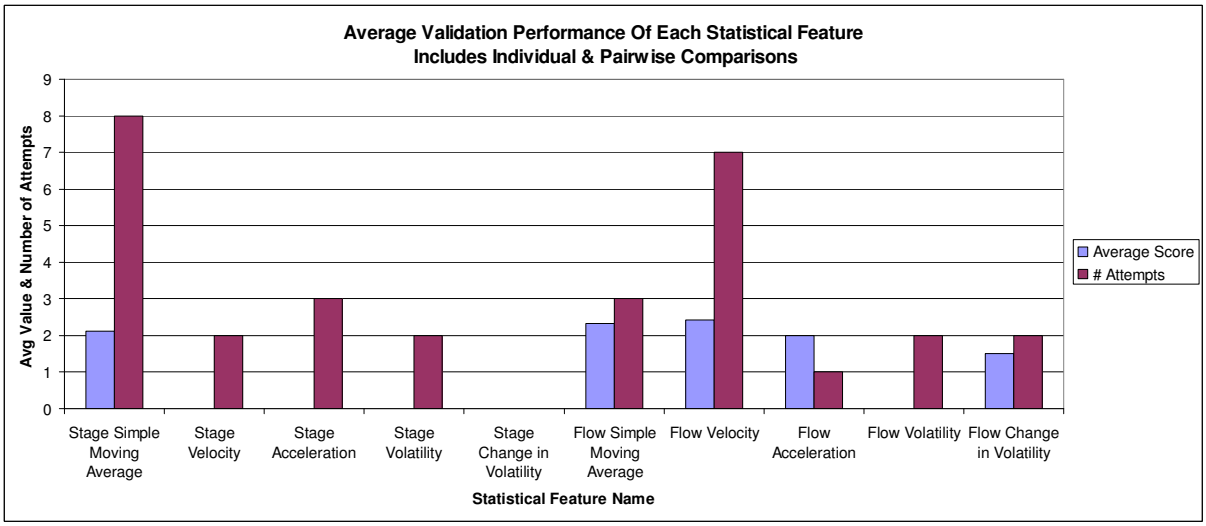

Figure 5.11. Summarised Results Given Successful Pre-Event Notification in Training Data

Chapter 6

Research Program

6.1 Discussion on Future Work

The initial results highlighted in Chapters 4 and 5 indicate this method has the potential to meet the demanding research goal and aims, and significant progress across the range of research milestones has been achieved. While most aspects of the "core" algorithm have already been developed, tested, and integrated, more work remains. A discussion of future work outside this research grant necessary to meet the seven objectives identified in Section 1.4 follows.

- *Objective 1. Develop easy to use, continuous data collection and management process for one or more data streams within an intelligent agent construct*

  Much progress has been made relating to Objective 1, but more work remains for the objective to be fully realised. An event monitoring capability has been successfully developed using the method identified in Figure 3.2. Data from multiple data streams can be collected by the data collection agent, read and processed by the data monitoring agent, and near real-time event monitoring can be performed using statistical feature settings identified by the data analysis agent. Additionally, event warning alerts can be displayed on the host terminal and automatically generated alert messages can be transmitted to other users at other terminals.

  As mentioned earlier, the currently implemented algorithm has the ability to manage and process more than one data stream at a time. The modified SPC tool (for combined feature generation and pattern recognition) represents the first stage of a two-stage pattern recognition process. In the first stage, each optimal statistical feature (and associated event warning limits) is individually or jointly developed for one data stream or across any two data streams in a pairwise fashion. The second stage of pattern recognition is performed by a neural network. It is in the second stage where all the optimized statistical features for one or more data streams relating to a single monitored process are combined to accomplish composite temporal pattern recognition. While effort has been placed on understanding some of the intricacies of neural networks, and in fact a basic neural network tool has already been developed, actual implementation and tight integration within the algorithm has not yet been accomplished.

  Other areas relating to Objective 1 still needing work include developing and improving the logic for controlling each component of the pattern recognition process in an attempt to ensure robust results, and the creation of an "agent health" monitoring capability. Computer network problems and the difficulties caused by network failures on the data collection agent underscore the need for a "health" monitoring agent capable of determining whether or not each agent is functioning properly and shutting down and restarting the agents and applications as necessary.

- *Objective 2. Develop method to generate highly tailored robust statistical features from time series data that best capture trend and volatility movement prior to and during previous significant events, capable of overcoming difficulties caused by non-stationary data*

  The single and dual-layered genetic algorithm approach appears to properly perform the combinatorial matching between the statistical feature settings and associated event warning limit settings. The direct evaluation of joint conditional relationships in the first stage of the pattern recognition process shows potential to further enhance the method's effectiveness. However, certain aspects of method performance, such as the type of mutation and cross over operators to employ, still need further analysis.

Another area yet to be explored includes pre-processing the data prior to building the statistical features. Disparate smoothing techniques commonly used in time series analyses can be employed to see if they improve the method's performance. Other options to explore include using optimally processed statistical feature data as the "raw" data source for altogether different statistical feature calculations. For example, Chapter 4 identified the value of the velocity statistical feature, producing an optimal data set of velocity metrics. This data could become the raw data for use by the other statistical features, perhaps enhancing visibility of the predicted event.

- *Objective 3. Develop method for monitoring, controlling, and configuring first-stage modified SPC tool*

  This objective relates to Objective 1 and the use of intelligent software agents to control the analysis process. While the methodology attempts to minimize the number of settings, preliminary sensitivity analysis results indicate changing some settings have greater affect than changing others. Accordingly, additional sensitivity analyses should be performed to identify robust initial settings and understand when, and to what degree, they should change throughout the analysis.

- *Objective 4. Develop method for combining statistical features from multiple data streams, generating a composite "snapshot" of the temporal process*

  This objective addresses the presentation of numerous statistical features to the neural network. The preliminary results support the notion that statistical features producing poor results with training data in the first stage pattern recognition process also tend to produce poor results in test and validation data. This is particularly true when comparing the number of time periods associated with correct pre-event notification between the training and test or validation data. Poor performing statistical features will be discarded prior to the second stage pattern recognition process, and further statistical feature reduction will occur in the second stage using the feature saliency screening method described in [11]. The preliminary results suggest statistical features capturing (and in particular predicting) event-related data pattern movement with the training data appear to offer the strongest potential for correctly identifying event-related data pattern movement in the test data, an intuitively appealing result.

- *Objective 5. Develop method to adaptively configure second-stage pattern recognition tool (a neural network)*

  This objective addresses how to configure the neural network to best perform the pattern recognition task from the given set of statistical features. Neural networks are prevalent across many fields of research because they offer such a wide range of architectures, configurations, and settings. Unfortunately there is little consensus within the literature for how to best match these variables to a given problem, and to date little work has been conducted in this area inside this research effort. As a result, two different types of preliminary sensitivity analyses still need to be performed. The first one must evaluate the different types of neural network architectures to identify which type best meets the goals of this research, whereas the second one must consider how to best configure the settings of the chosen architecture to this research problem.

- *Objective 6. Develop method for intelligent agent to interactively control overall (pre-) event detection analytical process*

  As mentioned in Objective 1, work has progressed significantly in this area but more work remains. As a better understanding of the method emerges through various sensitivity analyses, more insight will be obtained for how to best control the analysis process. Intelligent agents will ultimately be responsible for starting, stopping, and monitoring each part of the analysis. They

will be capable of changing the settings identified through the work accomplished within Objectives 3 and 5, and it will also need to interact with the user. The user, in turn, will be able to override the agent's choice of settings, as well as monitor the analysis progress through summary reports and descriptive event log files generated throughout the analysis process.

- *Objective 7. Perform preliminary sensitivity analyses on methodology using synthetic and real data*

  In order to evaluate the robustness of the methodology, several sensitivity analyses have been performed but more are required. The results in Chapters 4 and 5 indicate the approach is capable of identifying and possibly even predicting events despite being confronted with challenging and complex data. More investigation is warranted to better understand the capabilities and limitations of this approach. Careful examination of the results will likely reveal how some identified limitations may be reduced or eliminated through changes to the algorithms. Obtaining additional multi-channelled real data in different fields of research is critical to evaluating how the method performs in practice. Comparing these results against results from other researchers will prove insightful and will ultimately indicate whether or not future research efforts using this approach are justified.

Chapter 7

Conclusions

The Surrey Space Centre is at the forefront of a broad movement to make space more responsive, accessible, and affordable to an increasing number of users. Key elements in this overall strategy include improved space system utilization, increased operational efficiencies, and superior methods to analyze immense volumes of data. Developing and integrating innovative uses of autonomy across the whole spectrum of space-related activities, from space service providers to data consumers, is vital for SSC to meet its lofty goals.

As the primary developer of the DMC, the SSC is now investigating how to best monitor space and/or terrestrial source data streams for identifying interest-event occurrences. This research addresses the development of a novel method for enhanced (pre-) event detection, particularly when working with one or more complex temporal data streams such as those found in environmental monitoring. As Zimmerman noted when conducting water pollution monitoring [107], environmental data often shows evidence of seasonal variations, high natural variability, lack of independence, covariate effects, non-normal distributions, and auto correlation. While many methods exist to analyse time series data, rarely can they operate directly on data exhibiting these difficulties. To analyse such data using traditional time series analysis methods, if possible at all, requires meeting numerous data assumptions obtained following lengthy data transformation efforts frequently accomplished in trial-and-error fashion. Even if the data transformations are successful, traditional time series methods are not designed to perform pattern recognition tasks directly, leaving the researcher to serially explore numerous pattern recognition techniques. In summary, successfully accomplishing event detection with complex temporal data ultimately results in significant effort requiring substantial domain-specific expertise.

In contrast, the proposed method coherently employs a dual-layered optimization and pattern recognition approach controlled by intelligent software agents to overcome these challenges. In the first layer, by exploiting the unique combinatorial search benefits of genetic algorithms within a modified SPC construct, robust statistical features tailored to the temporal data of one or more data streams are calculated without first requiring the raw data to meet strict assumptions of normality and stationarity. In addition, by simultaneously solving for the features and a range of flexible event alert control limits, the proposed method optimally captures and characterises how the time series evolved prior to and during previous significant events. In the second layer, a composite temporal snapshot of the monitored process is developed and analysed by combining salient optimally tuned statistical features from one or more data streams within a tailored neural network. The author was unable to locate literature in any field of research where these key features had been previously combined.

A novel method was developed and evaluated over two different types of data sets. The first feasibility study contained two synthetic single channelled data sets with noisy, highly non-linear, non-stationary mean and variance data. The results reveal the method was able to readily detect the event signal despite the considerably low 1.5 signal-to-noise ratio. In the case where the event signal magnitude was reduced to equal the noise level (a 1.0 signal-to-noise level), the results indicated the method was still able to accurately detect the signal for the training and test data sets. In fact, using the velocity statistical feature for the data set with signal-to-noise ratio of 1.0, the method achieved an overall 99.8% prediction accuracy over 2000 test data points, correctly predicting each of the four events *before* they occurred (with an average warning interval of five time periods) while only generating a 0.2% false alarm rate. A comparison result using traditional time series methods and several neural networks was difficult to accomplish, and produced unsatisfying results. The best results only captured one event and produced no false alarms. The overall difficulty of analysing complex time series data for pattern recognition using other methods accentuates the benefits of the proposed approach. In this example, no data transformations were applied, and the feature

development and pattern recognition analysis were performed simultaneously by the genetic algorithm in one step.

The second feasibility study contained two channelled real river data obtained over a 20-year period. Despite the low event count in the data, the study demonstrated the method could successfully fuse two data streams through pairwise evaluations of all joint conditional relationships, an idea easily extended for more than two channelled data. With better training data, stronger conclusions could be drawn.

The final results from this research are intended not only to demonstrate the method's utility within the environmental monitoring field, but for the general problem of (pre-) event detection using temporal data. Significant progress across the range of research milestones and objectives has been achieved. Preliminary analysis results using single and two channelled data suggest the method is capable of identifying and perhaps useful in predicting complex event-related data patterns. These results strengthen our conviction in the potential of the method for successful (pre-) event detection on complex temporal data.

Way Forward

Significant progress has been achieved through the course of this research, and we believe the method offers the potential for tremendous benefit to a wide range of organisations interested in event detection and prediction using temporal data. Accordingly, we have outlined in Chapter 6 a number of research opportunities that if accomplished, would advance the method as well as develop a better appreciation for its application and potential. Researching these various activities will provide considerable insight into this method as well as likely prove insightful into a number of different research areas touching on each topic. A brief overview of the work and the benefits for each task are outlined below.

The first activity we would research involves tightly integrating a neural network into the pattern recognition process, creating a coherent method capable of simultaneous multi-channelled data fusion. The existing first layer pattern recognition capability offers a way to generate optimal input data by evaluating single and joint conditional relationships, however a neural network is needed to bring all the different relationships together at one time for a comprehensive assessment.

The second research activity generates the logic for how to interact with and control each stage of the pattern recognition process. Different techniques will being considered, with the result being a coherent logic flow for adjusting analysis parameters during training and monitoring activities to ensure optimal performance with robust statistical metrics.

The third research activity involves evaluating the method performance using different types of mutation and cross over operators within the two tiered genetic algorithm structure. With numerous operators and configurations to choose from, understanding how each one affects method performance is critical.

The fourth area of research will look at creating an “agent health” monitoring capability. This research will investigate how to monitor and interact with the other agents and applications. As the method grows in capacity and complexity, the need to ensure each component is operating properly becomes more and more important.

Bibliography

[1] Al-Assaf, Y. Recognition of control chart patterns using multi-resolution wavelet analysis and neural networks. *Computers & Industrial Engineering 47*: 17-29, 2004.

[2] Al-Ghanim, A. An unsupervised learning neural algorithm for identifying process behaviour on control charts and a comparison with supervised learning approaches. *Computers ind. Engng Vol. 32, No 3*: 627-639, 1997.

[3] Anctil, F., Michel, Claude., Perrin, C., et al. A soil moisture index as an auxiliary ANN input for stream flow forecasting. *Journal of Hydrology 286*: 155-167, 2004.

[4] Antunes, M., Turkman, A.A., Turkman, K.F. A Bayesian approach to event prediction. *Journal of Time Series Analysis 24(6)*: 631-646, 2003.

[5] Atiya, A.F., El-Shoura, S.M., Shaheen, S.I., El-Sherif, M.S. A comparison between neural-network forecasting techniques- Case study: River flow forecasting. *IEEE Transactions on Neural Networks 10(2)*: 402-409, 1999.

[6] Audenino, A.L., Belingardi, G. Processing of simultaneous mechanical random response signals: integration, differentiation and phase shifts correction. *Mechanical Systems and Signal Processing 10(3)*: 277-291, 1996.

[7] Babovic, V., Sannasiraj, S.A., Chan, E.S. Error correction of a predictive ocean wave model using local model approximation. *Journal of Marine Systems 53*: 1-17, 2005.

[8] Bagchi, Tapan P. *Multiobjective Scheduling by Genetic Algorithms*. Kluwer Academic Publishers, Boston, MA, 1999.

[9] Barbounis, T.G., Theocharis, J.B. Locally recurrent neural networks for long-term wind speed and power projection. *Neurocomputing*, to appear, 2005.

[10] Bate, R.R., Mueller, D.D., White, J.E. *Fundamentals of Astrodynamics*. Dover Publications, Inc., New York, 1971.

[11] Bauer, K.W., et al. Feature screening using signal-to-noise ratios. *Neurocomputing 31*: 29-44, 1999.

[12] Belue, L.M., Bauer, K.W. Determining input features for multilayer perceptrons, *Neurocomputing 7*: 111-121, 1995.

[13] Bishop, Christopher M. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1998.

[14] Blattner, P., Ulrich, L. *Using Microsoft Excel 2000, Special Edition*. Que Corporation, Indianaplois, IN, 1996.

[15] Boden, D.G., Larson, W.J. *Cost Effective Space Mission Operations*. McGraw-Hill Higher Education, 1996.

[16] Brahim-Belhouari, S., Bermak, A. Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis 47*: 705-712, 2004.

[17] BuHamra, S., Smaoui, N., Gabr, M. The Box-Jenkins analysis and neural networks: prediction and time series modelling. *Applied Mathematics Modelling 27*: 805-815, 2003.

[18] Burden, R.L., Faires, J.D. *Numerical Analysis, 6th Edition*. Brooks/Cole Publishing Company, Pacific Grove, CA, 1997.

[19] Burgess, A.N., Refenes, A-P.N. Modelling non-linear moving average processes using neural networks with error feedback: An application to implied volatility forecasting. *Signal Processing 74*: 89-99, 1999.

[20] Camps-Valls, G., Martinez-Ramon, M., et al. (2004). Robust γ filter using support vector machines. *Neurocomputing 62*, 493-499.

[21] Caporale, G.M., Cipollini, A., Demetriades, P.O. Monetary Policy and the exchange rate during the Asian crisis: identification through heteroscedasticity. *Journal of International Money and Finance 24*: 39-53, 2005.

[22] Chan, T.K.Y., Yan, E.C., Naralalka, N. *A novel neural network for data mining*. Proc. 8th Int. Conf. on Neural Information Processing (ICONIP2001), 2001, China.

[23] Chen, G., McAvoy, T.J. Predictive on-line monitoring of continuous processes. *J. Proc. Cont. 5(6)*: 409-420, 1998.

[24] Chen, W., Meer, P., Georgescu, B., He, W., et al. Image mining for investigative pathology using optimized feature extraction and data fusion. *Computer Methods and Programs in Biomedicine*, to appear, 2005.

[25] Cheng, C.S. (1995). A multi-layer neural network model for detecting changes in the process mean. *Computers ind. Engng 28*, 51-61.

[26] Coca, D., Billings, S.A. A direct approach to identification of non-linear differential models from discrete data. *Mechanical Systems and Signal Processing 13(5)*: 739-755, 1999.

[27] Coley, David A., *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific, Singapore, 2003.

[28] Corbett, C.J., Pan, JN. Evaluating environmental performance using statistical process control techniques. *European Journal of Operational Research 139*: 68-83, 2002.

[29] Currenti, G., Del Negro, C., Lapenna, V., Telesco, L. Fluctuation analysis of the hourly time variability of volcano-magnetic signals recorded at Mt. Etna Volcano, Sicily (Italy). *Chaos Solitons & Fractals 23*: 1921-1929, 2005.

[30] Curtis, Myron. Computer programs for obtaining kinetic data on human movement. *J. Biomechanics 1*: 221-234, 1968.

[31] De Magalhaes, M.S., Epprecht, E.K., Costa, A.F.B., Economic design of a Vp $\overline{X}$ chart. *Int. J. Production Economics 74*: 191-200, 2001.

[32] Dierckx, P. An algorithm for smoothing, differentiation and integration of experimental data using spline functions. *Journal of Computational and Applied Mathematics 1(3)*: 165-183, 1975.

[33] Dillon, W.R., Goldstein, M., *Multivariate Analysis Methods and Applications*. John Wiley & Sons, New York, 1984.

[34] Gazzani, Fabio. Comparative assessment of some algorithms for differentiating noisy biomechanical data. *International Journal of Bio-Medical Computing 37*: 57-76, 1994.

[35] Ge, X., Smyth P. *Deformable Markov model templates for time-series pattern matching*. In Proc. 6th Int. Conf. on Knowledge Discovery and Data Mining, Boston, MA, 2000.

[36] Gen, M., Cheng, R., *Genetic Algorithms & Engineering Design*. John Wiley & Sons, New York, 1997.

[37] Goldberg, David E., *Genetic Algorithms in Search, Optimization, & Machine Learning*. Addison-Wesley, Boston, MA, 2003.

[38] Ghiassi, M., Saidane, H., Zimbra, D.K. A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting 21*: 341-362, 2005.

[39] Guh, R.S. A hybrid learning-based model for on-line detection and analysis of control chart patterns. *Computers & Industrial Engineering 49*: 35-62, 2005.

[40] Guh, R.S. Robustness of the neural network based control chart pattern recognition system to non-normality. *International Journal of Quality & Reliability Management 19*: 97-112, 2001.

[41] Guh, R.S., Zorriassatine, F., Tannock, J.D.T., O'Brien, C. IntelliSPC: a hybrid intelligent tool for on-line economical statistical process control. *Expert Systems with Applications 17*: 195-212, 1999.

[42] Guh, R.S., Zorriassatine, F., Tannock, J.D.T., O'Brien, C. On-line control chart pattern detection and discrimination- a neural network approach. *Artificial Intelligence in Engineering 13*: 413-425, 1999.

[43] Guralnik, V., Sirvastava, J. *Event detection from time series data*. In Proc. 5th Int. Conf. on Knowledge Discovery and Data Mining, San Diego, CA, USA, 1999.

[44] Hauck, D. J., Runger, G. C., and Montgomery, D. C. (1999), "Multivariate Statistical Process Monitoring and Diagnosis with Grouped Regression-Adjusted Variables", *Communications in Statistics: Simulation and Computation*, Vol. 28, No. 2.

[45] Ipek, H., Ankara, H., Ozdag, H. The application of statistical process control. *Minerals Engineering 12(7)*: 827-835, 1999.

[46] Jiji, R.D., Hammond, M.H., Williams, F.W., Rose-Pehrsson, S.L. Multivariate statistical process control for continuous monitoring of networked early warning fire detection (FWFD) systems. *Sensors and Actuators B 93*: 107-116, 2003.

[47] Keogh, E.J., Pazzani, M.J. *Scaling up dynamic time warping for datamining applications*. In Proc. 6th Int. Conf. on Knowledge Discovery and Data Mining, Boston, MA, 2000.

[48] Kernighan, B.W., Ritchie, D.M. *The C Programming Language, 2nd Edition*. Prentice Hall PTR, New Jersey, 1988.

[49] Kim, T.Y., Oh, K.J., Kim, C., Do. J.D. Artificial neural networks for non-stationary time series. *Neurocomputing 61*: 439-447, 2004.

[50] Kuehl, Robert O. *Design of Experiments: Statistical Principles of Research Design and Analysis, 2nd Edition*. Duxbury Press, Pacific Grove, CA, 2000.

[51] Lanshammar, Hakan. On precision limits for derivatives numerically calculated from noisy data. *J. Biomechanics 15(6)*: 459-470, 1982.

[52] Larson, W.J., Wertz, J.R. *Space Mission Analysis and Design, 3rd Edition*. Microcosm Inc, Torrance, CA, 2003.

[53] Laskaris, N., Fotopoulos, S., Papathanasopoulos, P., Bezerianos, A. Robust moving averages, with Hopfield neural network implementation for monitoring evoked potential signals. *Electroencephalography and Clinical Neurophysiology 104*: 151-156, 1997.

[54] Lee, S., Choi, S. Adaptive process monitoring using scale CUSUM for serially correlated processes. *Computers ind. Engng Vol. 33, Nos 3-4*: 737-740, 1997.

[55] Lendasse, A., Francois, D., et al. Vector quantization: a weighted version for time-series forecasting. *Future Generation Computer Systems*, to appear 2005.

[56] Lin Y.C., Chao, Y.C. On the design of variable sample size and sampling intervals $\overline{X}$ charts under non-normality. *Int. J. Production Economics 96*: 249-261, 2005.

[57] Liu, H., Brown, D.E. A new point process transition density model for space-time event prediction. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 34(3)*:310-324, 2004.

[58] Lopes, J.A., Menezes, J.C. Multivariate monitoring of fermentation processes with non-linear modelling methods. *Analytica Chimica Acta 515*: 101-108, 2004.

[59] Ma, J., Perkins, S. *Online novelty detection on temporal sequences*. In Proc. 9th Int. Conf. on Knowledge Discovery and Data Mining, Washington, DC, USA, 2003.

[60] Martin, E.B., Morris, A.J. Non-parametric confidence bounds for process performance monitoring charts. *J. Proc. Cont. 6(6)*: 349-358, 1996.

[61] McCabe, M.F., Franks, S.W., Kalma, J.D. Calibration of a land surface model using multiple data sets. *Journal of Hydrology 302*: 209-222, 2005.

[62] Meijer, Erik. Matrix algebra for higher order moments. *Linear Algebra and its Applications*, to appear, 2005.

[63] Mora-Lopez, L., Mora, J., Morales-Bueno, R., et al. Modelling time series of climatic parameters with probabilistic finite automata. *Environmental Modelling & Software 20*: 753-760, 2005.

[64] Myers, R.H., Montgomery, D.C. *Response Surface Methodology: Process & Product Optimization Using Designed Experiments*. John Wiley & Sons, New York, 1995.

[65] Nanopoulos, A., Alcock, R., Manolopoulos, Y. Feature-based classification of time series data. *International Journal of Computer Research*: 49-61, 2001.

[66] Negnevitsky, Michael. *Artificial Intelligence: A Guide to Intelligent Systems*. Addison Wesley, Harlow, England, 2002.

[67] Neter, Kutner, Nachtsheim, Wasserman. *Applied Linear Statistical Models, 4th Edition*. McGraw-Hill, Boston, MA, 1996.

[68] Nunnari, G., Dorling, S., Schlink, U., et al. Modelling SO_2 concentration at a point with statistical approaches. *Environmental Modelling & Software 19*: 887-905, 2004.

[69] Parker, G.D. Timing patterns in event lists: Recurrent geomagnetic storms. *Journal of Atmospheric and Solar-Terrestrial Physics*, to appear, 2005.

[70] Pindyck, R.S., Rubinfeld, D.L. *Econometric Models & Economic Forecasts, 3rd Edition*. McGraw-Hill Inc., New York, 1991.

[71] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 2nd Edition, 1992.

[72] Principe, J.C., Euliano, N.R., Lefebvre, W.C. *Neural and Adaptive Systems*. John Wiley & Sons, Inc. New York, 2000.

[73] Qu, Y., Wang, C., Wang, X. *Supporting fast search in time series for movement patterns in multiple scales*. In Proc. 7th Int. Conf. on Information and Knowledge Management, Bethesda, Maryland, USA, 1998.

[74] Reeves, C.R., Rowe, J.E., *Genetic Algorithms-Principles and Perspectives: A Guide to GA Theory*. Kluwer Academic Publishers, Boston, MA, 2003.

[75] Ritter, Klaus. Almost optimal differentiation using noisy data. *Journal of Approximation Theory 86*: 293-309, 1996.

[76] Rius, A., Ruisanchez, I., et al. (1998). Reliability of analytical systems: use of control charts, time series models and recurrent neural networks (RNN). *Chemometrics and Intelligent Laboratory Systems 40*, 1-18.

[77] Roman, Steven. *Writing Excel Macros with VBA, 2nd Edition*. O'Reilly & Associates, Sebastopol, CA, 2002.

[78] Rose-Pehrsson, S., Shaffer, R.E., Hart., S.J., et al. Multi-criteria fire detection systems using a probabilistic neural network. *Sensors and Actuators B 69*: 325-335, 2000.

[79] Russell, S., Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall International, Inc. Upper Saddle River, NJ, 1995.

[80] Sall, J., Lehman, A. *JMP Start Statistics*. Duxbury Press, Belmont, CA, 1996.

[81] Sellers, Jerry J. *Understanding Space: An Introduction to Astrodynamics, 2nd Edition*. McGraw-Hill Companies, Inc. Boston, MA, 2004.

[82] Sentana, E., Fiorentini, G. Identification, estimation, and testing of conditionally heteroskedastic factor models. *Journal of Econometrics 102*: 143-164, 2001.

[83] Shao, X., Ma, C. A general approach to derivative calculation using wavelet transform. *Chemometrics and Intelligent Laboratory Systems 69*: 157-165, 2003.

[84] Soudan, K., Dierckx, P. Calculation of derivatives and Fourier coefficients of human motion data, which using spline functions. *J. Biomechanics 12*: 21-26, 1979.

[85] Stroustrup, Bjarne. *The C++ Programming Language, 3rd Edition*. Addison-Wesley, Reading, Massachusetts, 1997.

[86] Swain, A.K., Billings, S.A. Weighted complex orthogonal estimator for identifying linear and non-linear continuous time models from generalized frequency response functions. *Mechanical Systems and Signal Processing 12(2)*: 269-292, 1998.

[87] Tagaras, George. Dynamic control charts for finite production runs. *European Journal of Operational Research 91*: 38-55, 1996.

[88] Thissen, U., Melssen, W.J., Buydens, L.M.C. Nonlinear process monitoring using bottle-neck neural networks. *Analytica Chimica Acta 446*: 371-383, 2001.

[89] Toth, E., Brath, A., Montanari, A. Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of Hydrology 239*: 132-147, 2000.

[90] Udechukwu, A., Barker, K., Alhajj, R., *Discovering All Frequent Trends in Time Series*, In Proc. of Winter International Symposium on Information and Communication Technologies, ACM International Conference Proceedings, January 2004.

[91] Van Bellegem, S., Sachs, R. Forecasting economic time series with unconditional time-varying variance. *International Journal of Forecasting 20*: 611-627, 2004.

[92] Wackerly, Mendenhall, Scheaffer. *Mathematical Statistics with Applications, 5th Edition*. Duxbury Press, Belmont, CA, 1996.

[93] Wang, Q., Tenhunen, J., Dinh, N.Q., et al. Similarities in ground- and satellite-based NDVI time series and their relationship to physiological activity of a Scots pine forest in Finland. *Remote Sensing of Environment 93*: 225-237, 2004.

[94] Wang, W., Lu, W., Wang, X., Leung, A.Y.T. Prediction of maximum daily ozone level using combined neural network and statistical characteristics. *Environment International 29*: 555-562, 2003.

[95] Webb, Jeff. *Using Excel Visual Basic for Applications, Special 2nd Edition*. Que Corporation, Indianaplois, IN, 1996.

[96] Westerhuis, J.A., Gurden, S.P., Smilde, A.K. Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems 51*: 95-114, 2000.

[97] Winston, Wayne L. *Operations Research Applications and Algorithms, 4th Edition*. Brooks/Cole-Thomson Learning, Belmont, CA, 2004.

[98] Winter, D.A., Sidwall, H.G., Hobson, D.A. Measurement and reduction of noise in kinematics of locomotion. *J. Biomechanics 7*: 157-159, 1974.

[99] Wise, B.M., Gallagher, N.B. The process chemometrics approach to process monitoring and fault detection. *J. Process Control Vol. 6. No. 6*: 329-348, 1996.

[100] Wise, B.M., Gallagher, N.B., Butler, S.W., et al. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *J. Chemometrics 13*: 379-396, 1999.

[101] Wu, H., Salzberg, B., Donghui, Z. *Online event-driven subsequence matching over financial data streams*. Proc. of ACM SIGMOD Int'l Conf. on Management of Data, France, 2004.

[102] Yang, J.H., Yang, M.S. A control chart pattern recognition system using a statistical correlation coefficient method. *Computers & Industrial Engineering 48*: 205-221, 2005.

[103] Zhang, G.P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing 50*: 159-175, 2003.

[104] Zhang, G.P., Qi, M. Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research 160*: 501-514, 2005.

[105] Zhong, M., Lingras, P., Sharma, S. Estimation of missing traffic counts using factor, genetic, neural, and regression techniques. *Transportation Research Part C 12*: 139-166, 2004.

[106] Zhu, Y., Shasha, D. *Efficient elastic burst detection in data streams*. In Proc. 9th Int. Conf. on Knowledge Discovery and Data Mining, Washington, DC, USA, 2003.

[107] Zimmerman, S.M., Dardeau, M.R., Crozier, G.F., Wagstaff, B. The second battle of Mobile Bay – Using SPC to control the quality of water monitoring. *Computers ind. Engng 31(1)*: 257-260, 1996.

[108] Zorriassatine, F., Tannock, J.D.T., O'Brien, C. Using novelty detection to identify abnormalities caused by mean shifts in bivariate processes. *Computers & Industrial Engineering 44*: 385-408, 2003.

Appendix

The details contained in this Appendix represent one level of summarisation for the information discussed in Section 5.5. The tables are grouped one statistical feature per data stream at a time, and show the results when that feature is joined with all other combinations of statistical features across the two data streams in a pairwise fashion. The first ten tables only show those feature combinations found successful in producing pre-event notification within the training data set. It is from this group we would normally expect to find the best feature and setting combinations most likely able to predict future events. The second ten tables show all values regardless of their pre-event notification results. Water flow velocity appeared to perform particularly well joined with other water flow statistical features.

Water Stage Simple Moving Average				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None	16	1	16	1
Stage Simple Moving Average	22	7	27	10
Stage Velocity	16	1	11	1
Stage Acceleration	16	1	16	1
Stage Volatility	16	1	14	1
Stage Change in Volatility				
Flow Simple Moving Average	16	1	18	1
Flow Velocity	16	1	14	1
Flow Acceleration	19	5	-41	N/A
Flow Volatility	16	1	-1708	Did Not Detect
Flow Change in Volatility	16	1	15	1

Water Stage Velocity				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None				
Stage Simple Moving Average	16	1	11	1
Stage Velocity	5	2	-172	Did Not Detect
Stage Acceleration				
Stage Volatility	5	2	4	-1
Stage Change in Volatility				
Flow Simple Moving Average	18	4	-47	N/A
Flow Velocity	6	4	-10	N/A
Flow Acceleration				
Flow Volatility	11	7	-1	Did Not Detect
Flow Change in Volatility				

Water Stage Acceleration				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None				
Stage Simple Moving Average	16	1	16	1
Stage Velocity				
Stage Acceleration	1	3	0	0
Stage Volatility				
Stage Change in Volatility				
Flow Simple Moving Average	15	3	-46	N/A
Flow Velocity	9	5	2	-1
Flow Acceleration				
Flow Volatility				
Flow Change in Volatility				

Water Stage Volatility				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None				
Stage Simple Moving Average	16	1	14	1
Stage Velocity	5	2	4	-1
Stage Acceleration				
Stage Volatility				
Stage Change in Volatility				
Flow Simple Moving Average	10	2	-2	N/A
Flow Velocity	7	8	-16	N/A
Flow Acceleration				
Flow Volatility				
Flow Change in Volatility				

Water Stage Change in Volatility				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None				
Stage Simple Moving Average				
Stage Velocity	8	6	-187	Did Not Detect
Stage Acceleration				
Stage Volatility				
Stage Change in Volatility				
Flow Simple Moving Average				
Flow Velocity	9	4	-11	Did Not Detect
Flow Acceleration				
Flow Volatility				
Flow Change in Volatility				

Figure A.1. Water Stage Statistical Metrics Found Successful In Training Data

Water Flow Simple Moving Average				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None				
Stage Simple Moving Average	16	1	18	1
Stage Velocity	18	4	-47	N/A
Stage Acceleration	15	3	-46	N/A
Stage Volatility	10	2	-2	N/A
Stage Change in Volatility				
Flow Simple Moving Average	18	3	5	4
Flow Velocity	7	3	4	2
Flow Acceleration				
Flow Volatility				
Flow Change in Volatility				

Water Flow Velocity				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None	3	2	-1	Did Not Detect
Stage Simple Moving Average	16	1	14	1
Stage Velocity	6	4	-10	N/A
Stage Acceleration	9	5	2	-1
Stage Volatility	7	8	-16	N/A
Stage Change in Volatility	9	4	-11	Did Not Detect
Flow Simple Moving Average	16	2	1	4
Flow Velocity	8	4	6	3
Flow Acceleration	7	4	4	2
Flow Volatility	9	5	2	6
Flow Change in Volatility	7	3	0	2

Water Flow Acceleration				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None				
Stage Simple Moving Average	19	5	-41	N/A
Stage Velocity				
Stage Acceleration				
Stage Volatility				
Stage Change in Volatility				
Flow Simple Moving Average				
Flow Velocity	7	4	4	2
Flow Acceleration				
Flow Volatility				
Flow Change in Volatility				

Water Flow Volatility				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None				
Stage Simple Moving Average	16	1	-1708	Did Not Detect
Stage Velocity	11	7	-1	Did Not Detect
Stage Acceleration				
Stage Volatility				
Stage Change in Volatility				
Flow Simple Moving Average	16	1	-14	-9
Flow Velocity	11	7	3	9
Flow Acceleration				
Flow Volatility				
Flow Change in Volatility	11	4	-2	Did Not Detect

Water Flow Change in Volatility				
Joint Relationship Description	Training Score	# Training Pre-Event Periods	Validation Score	# Validation Pre-Event Periods
None	7	7	-19	Did Not Detect
Stage Simple Moving Average	16	1	15	1
Stage Velocity				
Stage Acceleration				
Stage Volatility				
Stage Change in Volatility				
Flow Simple Moving Average				
Flow Velocity	7	3	0	2
Flow Acceleration				
Flow Volatility	11	4	-2	Did Not Detect
Flow Change in Volatility	7	6	-21	Did Not Detect

Figure A.2. Water Flow Statistical Metrics Found Successful In Training Data

Water Stage Simple Moving Average

Joint Relationship Description	Training Results					Validation Results				
	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	16	25	100%	1	0	16	25	92%	1	2
Stage Simple Moving Average	22	33	100%	7	0	27	27	100%	10	0
Stage Velocity	16	18	100%	1	0	11	12	100%	1	0
Stage Acceleration	16	23	100%	1	0	16	23	100%	1	0
Stage Volatility	16	22	100%	1	0	14	26	85%	1	4
Stage Change in Volatility	15	23	100%	0	0	10	25	84%	0	4
Flow Simple Moving Average	16	24	100%	1	0	18	23	100%	1	0
Flow Velocity	16	19	100%	1	0	14	29	86%	1	4
Flow Acceleration	19	22	100%	5	0	-41	65	19%	N/A	N/A
Flow Volatility	16	24	100%	1	0	-1708	1728	1%	Did Not Detect	N/A
Flow Change in Volatility	16	24	100%	1	0	15	26	89%	1	3

Water Stage Velocity

Joint Relationship Description	Training Results					Validation Results				
	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Stage Simple Moving Average	16	18	100%	1	0	11	12	100%	1	0
Stage Velocity	5	5	100%	2	0	-172	172	0%	Did Not Detect	N/A
Stage Acceleration	5	5	100%	-7	0	4	4	100%	-9	0
Stage Volatility	5	5	100%	2	0	4	4	100%	-1	0
Stage Change in Volatility	-2	2	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Flow Simple Moving Average	18	25	96%	4	1	-47	81	25%	N/A	N/A
Flow Velocity	6	6	100%	4	0	-10	10	0%	N/A	N/A
Flow Acceleration	-1	1	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Flow Volatility	11	12	100%	7	0	-1	-1	0%	Did Not Detect	1
Flow Change in Volatility	3	3	100%	-4	0	-1616	1635	1%	Did Not Detect	N/A

Water Stage Acceleration

Joint Relationship Description	Training Results					Validation Results				
	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	2	2	100%	0	0	0	0	0%	Did Not Detect	N/A
Stage Simple Moving Average	16	23	100%	1	0	16	23	100%	1	0
Stage Velocity	5	5	100%	-7	0	4	4	100%	-9	0
Stage Acceleration	1	1	100%	3	0					
Stage Volatility	6	7	100%	-8	0	-6	12	25%	-10	9
Stage Change in Volatility	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Flow Simple Moving Average	15	16	100%	3	0	-46	54	7%	N/A	N/A
Flow Velocity	9	9	100%	5	0	2	6	67%	-1	2
Flow Acceleration	1	1	0%	-8	0	0	2	50%	-4	1
Flow Volatility	10	15	100%	-2	0	-393	393	0%	Did Not Detect	393
Flow Change in Volatility	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Water Stage Volatility

Joint Relationship Description	Training Results					Validation Results				
	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	0	1	100%	0	0	N/A	N/A	N/A	N/A	N/A
Stage Simple Moving Average	16	22	100%	1	0	14	26	85%	1	4
Stage Velocity	5	5	100%	2	0	4	4	100%	-1	0
Stage Acceleration	6	7	100%	-8	0	-6	12	25%	-10	9
Stage Volatility	0	1	100%	-6	0	N/A	N/A	N/A	N/A	N/A
Stage Change in Volatility	-2	2	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Flow Simple Moving Average	10	13	100%	2	0	-2	32	47%	N/A	N/A
Flow Velocity	7	7	100%	8	0	-16	34	27%	N/A	N/A
Flow Acceleration	5	5	100%	-7	0	-1	9	56%	N/A	N/A
Flow Volatility	4	10	1%	-1	0	-17	17	0%	Did Not Detect	17
Flow Change in Volatility	11	11	100%	-7	0	-12	12	0%	Did Not Detect	12

Water Stage Change in Volatility

Joint Relationship Description	Training Results					Validation Results				
	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Stage Simple Moving Average	15	23	100%	0	0	10	25	84%	0	4
Stage Velocity	8	8	100%	6	0	-187	187	0%	Did Not Detect	187
Stage Acceleration	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Stage Volatility	-2	2	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Stage Change in Volatility	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Flow Simple Moving Average	12	15	100%	-2	0	-5	5	0%	Did Not Detect	5
Flow Velocity	9	9	100%	4	0	-11	11	0%	Did Not Detect	11
Flow Acceleration	2	3	100%	-6	0	-2	6	33%	N/A	N/A
Flow Volatility	0	1	0%	-4	0	-2	2	0%	N/A	N/A
Flow Change in Volatility	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Figure A.3. Complete Summary of Water Stage Statistical Metrics

Water Flow Simple Moving Average	Training Results					Validation Results				
Joint Relationship Description	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Stage Simple Moving Average	16	24	100%	1	0	18	23	100%	1	0
Stage Velocity	18	25	96%	4	1	-47	81	25%	N/A	N/A
Stage Acceleration	15	16	100%	3	0	-46	54	7%	N/A	N/A
Stage Volatility	10	13	100%	2	0	-2	32	47%	N/A	N/A
Stage Change in Volatility	12	15	100%	-2	0	-5	5	0%	Did Not Detect	5
Flow Simple Moving Average	18	21	100%	3	0	5	5	100%	4	0
Flow Velocity	7	7	100%	3	0	4	14	64%	2	5
Flow Acceleration	14	14	100%	0	0	-3	3	0%	Did Not Detect	3
Flow Volatility	12	16	100%	-3	0	-12	30	30%	5	21
Flow Change in Volatility	12	23	100%	0	0	-6	6	0%	Did Not Detect	6

Water Flow Velocity	Training Results					Validation Results				
Joint Relationship Description	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	3	3	100%	2	0	-1	1	0%	Did Not Detect	1
Stage Simple Moving Average	16	19	100%	1	0	14	29	86%	1	4
Stage Velocity	6	6	100%	4	0	-10	10	0%	N/A	N/A
Stage Acceleration	9	9	100%	5	0	2	6	67%	-1	2
Stage Volatility	7	7	100%	8	0	-16	34	27%	N/A	N/A
Stage Change in Volatility	9	9	100%	4	0	-11	11	0%	Did Not Detect	11
Flow Simple Moving Average	16	16	100%	2	0	1	3	67%	4	1
Flow Velocity	8	8	100%	4	0	6	14	71%	3	4
Flow Acceleration	7	7	100%	4	0	4	10	70%	2	3
Flow Volatility	9	10	100%	5	0	2	8	63%	6	3
Flow Change in Volatility	7	7	100%	3	0	0	2	50%	2	1

Water Flow Acceleration	Training Results					Validation Results				
Joint Relationship Description	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	5	5	100%	-8	0	-1	9	44%	-10	4
Stage Simple Moving Average	19	22	100%	5	0	-41	65	19%	N/A	N/A
Stage Velocity	-1	1	0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Stage Acceleration	1	1	0%	-8	0	0	2	50%	-4	1
Stage Volatility	5	5	100%	-7	0	-1	9	56%	N/A	N/A
Stage Change in Volatility	2	3	100%	-6	0	-2	6	33%	N/A	N/A
Flow Simple Moving Average	14	14	100%	0	0	-3	3	0%	Did Not Detect	3
Flow Velocity	7	7	100%	4	0	4	10	70%	2	3
Flow Acceleration	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Flow Volatility	13	16	100%	0	0	-6	6	0%	Did Not Detect	6
Flow Change in Volatility										

Water Flow Volatility	Training Results					Validation Results				
Joint Relationship Description	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	11	11	100%	-7	0	-9	9	0%	Did Not Detect	9
Stage Simple Moving Average	16	24	100%	1	0	-1708	1728	1%	Did Not Detect	N/A
Stage Velocity	11	12	100%	7	0	-1	-1	0%	Did Not Detect	1
Stage Acceleration	10	15	100%	-2	0	-393	393	0%	Did Not Detect	393
Stage Volatility	4	10	1%	-1	0	-17	17	0%	Did Not Detect	17
Stage Change in Volatility	0	1	0%	-4	0	-2	2	0%	N/A	2
Flow Simple Moving Average	16	23	100%	1	0	-14	46	33%	-9	21
Flow Velocity	11	11	100%	7	0	3	11	64%	9	3
Flow Acceleration	13	16	100%	0	0	-6	6	0%	Did Not Detect	6
Flow Volatility	6	12	75%	-8	3	-27	27	0%	Did Not Detect	27
Flow Change in Volatility	11	14	100%	4	0	-2	2	0%	Did Not Detect	2

Water Flow Change in Volatility	Training Results					Validation Results				
Joint Relationship Description	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms	Score	# Limits Exceeded	Success Rate	# Pre-event Time Periods	# False Alarms
None	7	7	100%	7	0	-19	19	0%	Did Not Detect	19
Stage Simple Moving Average	16	24	100%	1	0	15	26	89%	1	3
Stage Velocity	3	3	100%	-4	0	-1616	1635	1%	Did Not Detect	N/A
Stage Acceleration	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Stage Volatility	11	11	100%	-7	0	-12	12	0%	Did Not Detect	12
Stage Change in Volatility	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Flow Simple Moving Average	12	23	100%	0	0	-6	6	0%	Did Not Detect	6
Flow Velocity	7	7	100%	3	0	0	2	50%	2	1
Flow Acceleration	7	7	100%			-20	20	0%	Did Not Detect	20
Flow Volatility	11	14	100%	4	0	-2	2	0%	Did Not Detect	2
Flow Change in Volatility	7	7	100%	6	0	-21	21	0%	Did Not Detect	21

Figure A.4. Complete Summary of Water Flow Statistical Metrics